

KOREAN PATENT ABSTRACT (KR)

PUBLICATION

(51) IPC Code: G11B 20/00

(11) Publication No.: P2000-0077406

(43) Publication Date: 26 December 2000

(21) Application No.: 10-2000-0028123

(22) Application Date: 24 May 2000

(71) Applicant:

Nobujuki Idei, Sony Corporation

6-7-35 Kitashinagawa Shibuya-ku, Tokyo, Japan

(72) Inventor:

Teruyuki Shidara et al.

(54) Title of the Invention:

Data storage medium, recording apparatus and method therefor, and editing apparatus and method therefor

Abstract:

One or more main data, management data for the main data, one or more auxiliary data related with the main data, and auxiliary management data for the auxiliary data are recorded in a data storage medium, and the auxiliary data and the auxiliary management data are linked to each other so that the auxiliary data and the auxiliary management data can be referred to based on the management data, and a data editing process can be simplified.

**Best Available Copy**

특 2000-0077406

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl. <sup>8</sup> G11B 20/00	(11) 공개번호 (43) 공개일자	특 2000-0077406 2000년 12월 26일
(21) 출원번호	10-2000-0028123	
(22) 출원일자	2000년 05월 24일	
(30) 우선권주장	99-144957 1999년 05월 25일 일본 (JP)	
(71) 출원인	소니 가부시끼 가이샤 미데이 노부유키	
(72) 발명자	일본국 도쿄도 시나가와쿠 키타시나가와 6초메 7반 35고 시타라테루유키 일본국도쿄도시나가와쿠키타시나가와6초메7반35고소니가부시끼가이샤내 야마다에이미치 일본국도쿄도시나가와쿠키타시나가와6초메7반35고소니가부시끼가이샤내 요코라테페이 일본국도쿄도시나가와쿠키타시나가와6초메7반35고소니가부시끼가이샤내 키하라노부유키 일본국도쿄도시나가와쿠키타시나가와6초메7반35고소니가부시끼가이샤내 신잔호	
(74) 대리인	신잔호	

심사청구 : 없음

(54) 기록매체, 기록장치, 기록방법, 편집장치 및 편집방법

요약

본 발명의 목적은 한 개 또는 다수의 메인데이터, 상기 메인데이터를 제어하는 관리데이터, 상기 메인데이터와 관련된 한 개 또는 다수의 보조데이터와 보조데이터를 제어하는 보조 관리데이터를 기록매체에 기록하고, 보조데이터와 보조 관리데이터가 관리데이터로부터 참조될 수 있도록 보조데이터와 보조 관리데이터를 연결시킴으로써 편집하는 작업을 단순화시키는 것이다.

도표도

도 8

명세서

도면의 간단한 설명

도 1a ~ 1d는 본 발명의 실시예로 제공된 메모리시트의 외형을 상면도, 정면도, 측면도 및 저면도로 각각 나타내는 도면이다.

도 2a는 실시예로 제공된 드라이브장치의 정면도, 상면도, 좌측면도, 우측면도, 저면도를 각각 나타내는 도면이다.

도 3은 드라이브장치의 내부 구조를 나타내는 블록도이다.

도 4는 실시예로 제공된 드라이브장치 및 메모리시트를 포함하는 전형적인 시스템 접속을 나타내는 설명도이다.

도 5는 실시예로 제공된 메모리시트의 파일시스템 처리개층을 나타내는 설명도이다.

도 6a ~ 6f는 실시예에 제공된 메모리시트의 데이터구조를 나타내는 설명도이며, 도 6a는 세그먼트로 명명된 데이터부를 나타내고, 도 6b는 사용자 블록을 나타내고, 도 6c는 파트를 나타내고, 도 6d는 파트의 잔여부분을 나타내고, 도 6e는 부트블록을 나타내고, 도 6f는 백업블록을 나타낸다.

도 7은 실시예에 제공된 메모리시트의 디렉토리구조를 나타내는 설명도이다.

도 8a, 도 8b 및 도 8c는 실시예에 제공된 메모리시트의 관리구조를 구성하는 TRKLIST.MSF와 A30(n).MSA 및 INFLIST.MSF를 각각 나타내는 설명도이다.

도 9a, 도 9b 및 도 9c는 실시예에 제공된 메모리시트에 저장된 데이터파일을 나타내는 설명도이며, 도 9a는 데이터파일 #1의 구조를 나타내고, 도 9b는 부분간의 경계부를 나타내고, 도 9c는 데이터파일 #2의 구조를 나타낸다.

도 10은 실시예에 제공된 메모리시트에 저장된 데이터파일을 결합하기 위한 조작을 나타내는 설명도이다.

다.

도 11a 및 도 11b는 실시예에 제공된 메모리시트에 저장된 데이터파일을 분류하기 위한 조작을 나타내는 각각의 데이터파일 #1과 데이터파일 #2의 설명도이다.

도 12a 및 도 12b는 설명도로서 도 12a는 실시예에 제공된 메모리시트에 저장된 트랙정보 관리파일의 데이터구조를 나타내고, 도 12b는 트랙정보 관리파일의 TRKINF/PRTINF의 세부 구조를 나타낸다.

도 13은 실시예에 제공된 트랙정보 관리파일 A-nnn의 하위 바이트를 나타내는 설명도이다.

도 14는 실시예에 제공된 트랙정보 관리파일 A-nnn의 상위 바이트를 나타내는 설명도이다.

도 15는 실시예에 제공된 트랙정보 관리파일의 NAME1의 데이터구조를 나타내는 설명도이다.

도 16은 실시예에 제공된 트랙정보 관리파일의 NAME2의 데이터구조를 나타내는 설명도이다.

도 17은 실시예에 제공된 데이터파일의 데이터구조를 나타내는 설명도이다.

도 18은 실시예에 제공된 메모리시트의 기록모드를 설명하는데 사용된 설명도이다.

도 19a 및 도 19b는 설명도로서 도 19a는 실시예에 제공된 부가정보파일의 데이터구조를 나타내고, 도 19b는 부가정보파일의 인덱스 포인터를 나타낸다.

도 20은 실시예에 제공된 부가정보유닛의 데이터구조를 나타내는 설명도이다.

도 21은 실시예에 제공된 전형적인 부가정보를 나타내는 설명도이다.

도 22는 실시예에 제공된 부가정보유닛을 나타내는 설명도이다.

도 23은 실시예에 제공된 전형적인 부가정보유닛을 나타내는 설명도이다.

도 24는 실시예에 제공된 다른 전형적인 부가정보유닛을 나타내는 설명도이다.

도 25는 실시예에 제공된 또 다른 전형적인 부가정보유닛을 나타내는 설명도이다.

도 26은 실시예에 제공된 부가정보의 삭제유닛의 예를 나타내는 설명도이다.

도 27은 파일을 기록하기 위해 실시예에 제공된 드라이브장치로 실행된 처리를 나타내는 플로우차트를 나타낸다.

도 28은 파일을 삭제하기 위해 실시예에 제공된 드라이브장치로 실행되는 처리를 나타내는 플로우차트를 나타낸다.

도 29는 실시예에 제공된 파일 압축처리를 나타내는 플로우차트를 나타낸다.

도 30a 및 도 30b는 실시예에 제공된 파일 압축처리 전후의 부가정보파일을 각각 나타내는 설명도이다.

\* 도면의 주요부분에 대한 부호설명

- |               |               |
|---------------|---------------|
| 1. 메모리시트      | 20. 드라이브장치    |
| 21. 표시부       | 22. 착탈기구      |
| 23. 헤드폰 출력단자  | 24. 라인출력단자    |
| 25. 마이크 입력단자  | 26. 라인입력단자    |
| 27. 디지털 입력단자  | 30. 조작부       |
| 31. 조작레벨      | 32. 정지키       |
| 33. 기록키       | 41. CPU       |
| 42. 메모리 인터페이스 | 43. USB 인터페이스 |
| 44. 리얼타임블록    | 45. 표시드라이브    |
| 48. 플래시메모리    | 49. DSP       |
| 50. SAM       |               |

## 발명의 상세한 설명

### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 하나 또는 복수의 메인데이터, 메인데이터를 제어하는 관리데이터, 메인데이터와 관련된 하나 또는 복수의 서브데이터 및 서브데이터를 제어하는 서브관리데이터용 기록매체와 기록장치 및 기록매체에 데이터를 기록하는 기록방법 및 편집장치 그리고 기록매체에 기록된 서브데이터의 편집을 실행하는 편집방법에 관한 것이다.

최근에 플래시메모리와 같은 고체기록소자를 탑재한 소형 기록매체내에 그리고 소형 기록매체로부터 전용 드라이브장치 또는 일반적인 드라이브장치를 이용하여 정지화상 데이터, 동화상 데이터, 음악데이터 및 오디오데이터 등의 각종 데이터를 기록 및 재생하는 오디오 또는 비디오장치 및 정보장치가 개발되고

있다.

이러한 기록매체에서는, 기록 및 재생조작이 장치마다 실행될 수 있다. 더욱이, 각종 데이터의 처리가 퍼스널 컴퓨터와 같은 정보장치 또는 CD플레이어, MO플레이어 및 비디오장치 등의 비디오 영상기기와 같은 다른 장치를 포함하는 시스템에서 실행될 수 있다.

이러한 고체기억소자를 이용하는 기억시스템에 저장된 파일을 편집하는 처리가 요구되고 있다.

더욱이, 요구된 편집 가능성 뿐만 아니라 더 효과적인 편집처리가 요구되고 있다. 예를 들면, 효과적인 편집처리에 있어서, 최소한의 편집처리시간과 최소한의 전력소비뿐만 아니라 기록매체에서 파일편집에 요구된 데이터의 전송, 복사 및 재감기와 같은 최소의 조작이 요구된다.

특히, 이러한 기록매체에 오디오 및 비디오데이터를 부가데이터와 관련시킴으로써 부가데이터를 기록하는 것이 고려될 수 있다. 부가데이터 자체는 상대적으로 큰 크기를 갖는다. 이러한 이유 때문에, 1개의 부가데이터의 삭제 등과 같은 편집작업을 실행하기를 원할 때 전체 부가데이터를 재감는 것과 같은 부가데이터를 처리하기 위해서 처리장치의 높은 처리능력이 또한 요구되고 있다.

더욱이, 이러한 처리가 예를 들면 상용 퍼스널 컴퓨터로 충분히 실행되는 경우라도, 어떤 경우에는 이러한 처리는 휴대용 기록/재생장치에 대해서 너무 큰 부담이 될 수 있다.

이러한 사정을 감안하여, 보다 효율적으로 그리고 적은 처리부담으로 편집작업이 실행될 수 있는 관리시스템이 제공되기를 원한다.

### 본 발명이 이루고자 하는 기술적 과제

상기 기술된 문제에 따라서, 본 발명은 부가정보의 제어 및 편집에 대해서 간단한 처리로 작업을 실행할 수 있는 관리시스템을 제공하는 것을 목적으로 한다.

본 발명의 제 1관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 한 개 또는 다수의 보조데이터를 기록하는 기록매체에 있어서,

상기 한 개 또는 다수의 메인데이터를 기록하는 메인데이터 기록영역과, 상기 한 개 또는 다수의 보조데이터로 구성된 보조데이터 파일 중 최소한 한 개의 파일을 기록하여 상기 보조데이터의 기록 위치를 제어하는 보조데이터 기록영역과, 상기 메인데이터 기록영역에 기록된 상기 메인데이터를 제어하고 상기 메인데이터를 상기 메인데이터와 관련된 상기 보조데이터를 포함하는 상기 보조데이터 파일에 링크(link)시키는 링크정보를 기록하는 관리데이터 기록영역으로 구성되는 것을 특징으로 한다.

본 발명의 제 2관점에 따르면, 한 개 또는 다수의 메인데이터, 상기 메인데이터와 관련된 한 개 또는 다수의 보조데이터와 상기 메인데이터를 제어하는 관리데이터를 기록하는 기록매체에 데이터를 기록하는 방법에 있어서,

상기 메인데이터를 상기 기록매체에 기록하는 단계와, 상기 메인데이터의 기록에 근거하여 상기 관리데이터를 기록하는 단계와, 상기 한 개 또는 다수의 보조데이터로 구성된 보조데이터 파일을 상기 기록매체내에 기록하는 단계와, 상기 메인데이터를 상기 보조데이터에 링크시키는 링크정보를 기록하기 위해 상기 관리데이터를 갱신하는 단계로 구성되는 것을 특징으로 한다.

본 발명의 제 3관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체상에 기록된 보조데이터의 무효데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집방법에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 데이터 판독단계와, 상기 관리데이터의 링크정보에 따라 상기 기록매체로부터 상기 보조데이터 파일을 판독하는 파일 판독단계와, 상기 보조데이터가 상기 보조데이터 파일내의 상기 보조 관리데이터에 근거하여 무효한 것이라고 판단되는 경우에, 상기 보조데이터 파일내의 상기 관리데이터에 의해 무효화된 보조데이터와 결합된 관리데이터를 삭제하고 상기 보조데이터 파일로부터 무효라고 판단된 상기 보조데이터를 삭제하는 삭제단계와, 상기 삭제 단계에서 삭제된 불필요한 정보를 가진 상기 보조데이터파일을 상기 기록매체내에 다시 입력시키는 파일 입력단계로 구성되는 것을 특징으로 한다.

본 발명의 제 4관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체상에 기록된 보조데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집방법에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 데이터 판독단계와, 상기 관리데이터의 링크정보에 따라 상기 기록매체로부터 상기 보조데이터 파일을 판독하는 파일 판독단계와, 상기 파일 판독단계에서 판독된 상기 보조데이터 파일내에서 삭제된다고 명시되어 있는 보조데이터와 결합된 특정한 한 개 또는 다수의 보조 관리데이터를 재입력시키는 재입력 단계와, 상기 재입력 단계에서 재입력된 상기 불필요한 보조데이터에 대한 명시된 한 개 또는 다수의 보조 관리데이터를 가진 상기 보조데이터 파일을 상기 기록매체내에 다시 입력시키는 파일 입력단계로 구성되는 것을 특징으로 한다.

본 발명의 제 5관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록매체에 기록하는 기록매체상에 기록장치에 있어서,

상기 메인데이터, 상기 관리데이터 또는 상기 보조 데이터 파일을 상기 기록매체에 기록하는 기록 수단과, 상기 메인데이터를 상기 기록매체에 기록한 사실에 근거하여 상기 기록매체상에 저장된 상기 관리

이터를 갱신하는 제 1 갱신수단과, 서브 데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 보조 관리데이터를 생성하고, 보조데이터 중 최소한 한 개를 제어하는 보조 관리데이터를 발생시키는 보조 데이터 파일 발생수단과, 상기 메인데이터를 가지는 상기 보조데이터 파일과 결합된 링크정보를 상기 관리 데이터 중 1개 데이터로서 기록하기 위해 상기 관리데이터를 갱신시키는 제 2 갱신수단으로 구성되는 것을 특징으로 한다.

본 발명의 제 6관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체에 기록된 보조데이터의 무효데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집장치에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 판독수단과, 상기 데이터를 상기 기록매체에 입력시키는 입력수단과, 상기 기록수단에 의해 상기 기록매체로부터 판독된 상기 보조데이터 파일내의 상기 보조 관리데이터에 근거하여 보조데이터 중 무효한 것이 있는지 또는 없는지를 판단하는 판단수단과, 상기 판단수단에 의해 형성된 상기 판단의 결과가 보조데이터 중 무효한 것이 있다고 지시하는 경우에 상기 보조데이터 파일을 편집하는 작업을, 상기 보조데이터 파일내에 있는 상기 보조 관리데이터에 의해 무효화된 보조데이터와 결합된 관리데이터를 삭제하는 단계와, 상기 보조데이터 파일로부터 무효라고 판정된 상기 보조데이터를 삭제하는 단계와, 상기 편집된 보조데이터 파일을 상기 기록매체에 입력시키는 상기 입력수단을 제어하는 단계를 실행시킴으로써 제어할 수 있는 제어수단으로 구성되는 것을 특징으로 한다.

본 발명의 제 7관점에 따르면, 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체에 기록된 보조데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집장치에 있어서,

상기 기록매체로부터 상기 데이터를 판독하고, 상기 기록매체에 입력시키는 액세스수단과, 상기 액세스수단에 의해 판독된 상기 보조데이터 파일내에서 삭제되는 보조데이터와 결합된 한 개 또는 다수의 보조 관리데이터를 재입력시키는 단계와, 상기 기록 매체내에 재입력된 상기 명시된 한 개 또는 다수의 보조 관리데이터를 가지는 상기 보조데이터 파일을 다시 입력시키는 단계를 실행시킴으로써 상기 액세스수단을 제어하는 제어수단으로 구성되는 것을 특징으로 한다.

#### 본 발명의 구성 및 작용

본 발명의 바람직한 실시의 형태에 대해서 다음과 같이 기술한다. 실시예에 제공된 기록매체, 즉 본 발명의 기록매체는 판형의 외형을 가지는 메모리인 메모리시트(memory sheet)를 예로서 설명한다.

본 발명의 데이터 처리장치는 메모리시트에 그리고 메모리시트로부터 데이터를 기록 및 재생할 수 있는 드라이브장치를 예로서 설명한다.

본 발명에 제공된 데이터파일, 관리파일 및 부가정보파일은 이후에 각각 기술되는 A30nnnnn.MSA로 명명된 데이터파일, TRKLIST.MSF로 명명된 정보관리파일 및 INFLIST.MSF로 명명된 부가정보파일로 예시될 수 있다.

설명은 아래의 순서로 나타난다.

1. 메모리시트
2. 드라이브장치의 구성
3. 전형적인 시스템 접속
4. 파일시스템
  - 4-1 처리구조 및 데이터구조
  - 4-2 디렉토리구조
  - 4-3 관리구조
  - 4-4 트랙정보 관리파일(TRKLIST.MSF)
  - 4-5 데이터파일(A30nnnnn.MSA)
  - 4-6 부가정보파일(INFLIST.MSF)
5. 파일기록처리
6. 파일삭제처리
7. 부가정보파일 압축처리

#### 1. 메모리시트

우선, 실시예에 제공된 기록매체로서 사용된 메모리시트(1)의 외형을 도 1a ~ 도 1d를 참조해서 설명한다.

메모리시트(1)는 도 1a ~ 도 1d에 나타난 것과 같은 판형 케이스에 수용된 전형적인 소정의 기억용량의 메모리소자를 갖는다. 본 실시예에서, 메모리소자는 플래시메모리이다.

도 1a ~ 도 1d는 플라스틱 몰드로 형성된 케이스의 상면도, 정면도, 측면도 및 저면도를 나타내는 도면이다.

일례로서, 도면에 나타난 것처럼 케이스는 60mm의 깊이(W11), 20mm의 폭(W12) 및 2.8mm의 두께(W13)를 갖는다.

케이스의 정면 저면측에는, 10개의 전극을 포함하는 단자부(2)가 형성된다. 데이터는 단자부(2)를 통해서 메모리소자에 그리고 메모리소자로부터 기입 및 독출된다.

케이스의 좌측 상부코너는 평면방향으로 향한 절결부(3)이다. 절결부(3)는 케이스가 착탈기구로 삽입되는 경우에 잘못된 방향으로 메모리시트(1)가 착탈기구에 장착되는 것을 방지한다.

라벨접착면(label sticking surface)(4)은 케이스의 상면 및 저면의 위치에 걸쳐서 설치된다. 사용자는 라벨 접착면(4)에 저장된 데이터를 기술하기 위한 라벨을 접착한다.

슬라이드 스위치(5)는 저면에 설치된다. 슬라이드 스위치(5)는 기록된 데이터가 부주위로 삭제되는 것을 방지한다.

메모리시트(1)에 사용된 플래시메모리의 기억용량은 4MB, 8MB, 16MB, 32MB, 64MB 또는 128MB로 규정되며, 기호 MB는 megabytes의 축약이다.

소위 FAT(File Allocation Table)시스템은 데이터를 기록 및 재생하기 위한 파일시스템으로 사용된다.

기입속도는 1,500kbytes/sec~330kbytes/sec이고, 독출속도는 2.45Mbytes/sec이다. 데이터는 512bytes단위로 기입되고, 8KB 또는 16KB 블록 단위로 삭제되고, 기호 KB는 kilobytes의 축약이다.

전원전압(Vcc)는 2.7V ~ 3.6V이며, 시리얼 클럭신호(SCLK)는 최대 20MHz의 주파수를 갖는다.

## 2. 드라이브장치와 구성

본 실시예에 제공되고 데이터를 메모리시트(1)에 그리고 메모리시트(1)로부터 기록 및 재생할 수 있는 드라이브장치(20)의 구성은 도 2a ~ 도 2e 및 도 3을 참조해서 설명한다.

드라이브장치(20)는 메모리시트(1)에 기입 및 메모리시트(1)로부터 독출되는 다수의 메인데이터의 종류를 처리할 수 있는 것에 주목해야 한다.

데이터의 일례는 동화상 데이터, 정지화상 데이터, 음성데이터, 오디오데이터, 음악데이터 및 제어데이터이다.

다음 기술에 있어서, 데이터는 주로 오디오데이터 또는 음악데이터로 가정한다.

도 2a ~ 도 2e는 드라이브장치(20)의 전형적인 외형의 정면도, 상면도, 좌측면도, 우측면도 및 저면도를 각각 나타낸다.

드라이브장치(20)는 소형 경량을 갖도록 고안되기 때문에, 사용자는 드라이브장치(20)를 용이하게 실행할 수 있다.

메모리시트(1)는 도 2b에 나타난 것처럼 드라이브장치(20)의 상부면에 설치된 착탈기구(22)에 장착된다. 음악데이터, 오디오데이터, 동화상 데이터, 정지화상 데이터, 컴퓨터 데이터 및 제어데이터와 같은 각종 데이터가 메모리시트(1)에 그리고 메모리시트(1)로부터 드라이브장치(20)에 의해 기록 및 재생된다.

액정패널로 사용된 표시부(21)는 드라이브장치(20)의 상부에 설치된다. 표시부(21)는 다른 사물 중 재생된 화상, 재생된 음성 또는 재생된 악곡과 관련된 정보, 사용자 조작의 가이드로서 사용되는 메시지와 재생 및 편집조작용 메뉴화면을 표시하는데 사용된다.

드라이브장치(20)는 이후에 기술되는 각종 기구와 드라이브장치(20)를 접속하기 위한 각종 단자를 또한 갖는다.

예를 들면, 헤드폰단자(23) 및 마이크로폰 입력단자(25)는 도 2b에 나타난 바와 같이 드라이브장치(20)의 상부에 설치된다.

헤드폰은 헤드폰단자(23)에 접속된다. 사용자는 재생 오디오신호로서 헤드폰으로 드라이브장치(20)에서 공급되는 재생된 음성을 청취할 수 있다.

마이크로폰은 마이크로폰 입력단자(25)에 접속된다. 드라이브장치(20)는 마이크로폰으로 픽업된 입력 오디오신호를 전형적인 메모리시트(1)에 기록할 수 있다.

라인출력단자(24), 라인입력단자(26) 및 디지털 입력단자(27)는 도 2c에 나타난 바와 같이 드라이브장치(20)의 우측에 설치된다.

재생 오디오신호는 오디오케이블로 라인출력단자에 접속된 외부장치에 공급될 수 있다. 예를 들면, 오디오증폭기를 라인출력단자(24)에 접속함으로써, 사용자는 스피커시스템을 통해서 메모리시트(1)로부터 재생된 악곡 및 음성을 청취할 수 있다. 선택적으로, 미니디스크(mini disc) 레코더 또는 테이프레코더를 라인출력단자(26)에 접속함으로써, 메모리시트(1)로부터 재생된 음악 또는 음성을 다른 매체에 더빙(dubbing) 및 기록할 수 있다.

더욱이, CD플레이어와 같은 외부장치는 라인입력단자(26)에 접속할 수 있다. 이 경우에, 외부장치에서 발생한 오디오신호는 메모리시트(1)에 전형적으로 기록되는 드라이브장치(20)에 공급될 수 있다.

또한, 드라이브장치(20)는 디지털 입력단자(27)에 접속된 광케이블을 통해서 디지털 소스장치로 전송되는 디지털 오디오데이터를 입력할 수 있다.

디지털 소스장치의 일예는 외부 CD플레이어이다. 이러한 외부 CD플레이어를 광케이블로 디지털 입력 단자(27)에 접속함으로써, 소위 디지털 더빙작업이 실행될 수 있다.

USB(Universal Serial Bus)커넥터(28) 및 전원단자(29)와 같은 단자는 도 2c에 나타낸 바와 같이 드라이브 장치(20)의 좌측에 설치된다.

USB커넥터(28)는 전형적인 USB인터페이스로 제공된 퍼스널 컴퓨터와 같은 USB기구에 접속된다. 이 경우에, 드라이브장치(20)는 USB기기로 각종 통신 및 데이터전송을 실행할 수 있다.

본 실시예에 제공된 드라이브장치(20)는 드라이브장치(20)를 구동하는 전원공급으로서 사용되도록 삽입된 건전지 또는 충전지를 사용한다. 그러나, 전원공급단자(29)는 드라이브장치(20)의 동작에 전원을 공급하기 위해 상용 교류전류에 전원어댑터로 접속될 수 있다.

상기 기술한 이들 단자의 종류, 단자의 수 및 상기 기술된 이들 위치는 단지 일예임에 주목해야 한다. 즉, 필요에 따른 많은 다른 종류의 단자는 상기 기술된 것과 다른 위치에 설치될 수 있다.

예를 들면, 광케이블용 디지털 출력단자가 설치될 수 있다. 다른 예로는 SCSI(Small Computer System Interface)커넥터, 시리얼포트, RS232C커넥터, IEEE1394(Institute of Electrical and Electronic Engineers 1394)커넥터 또는 다른 커넥터의 이용이다.

단자의 구조는 이미 공지되어 있기 때문에, 이들에 대한 기술은 생략한다. 그러나, 선택적인 구조로서, 단일 단자가 헤드폰단자(23) 및 라인출력단자(24)로서 사용될 수 있고, 또한 디지털 출력단자로서도 사용될 수 있다.

마찬가지로, 마이크로폰 입력단자(25), 라인입력단자(26) 및 디지털 입력단자(27)는 공용의 단일단자로 사용될 수 있다.

사용자에 의해 조작되는 조작레버(31)에 설치되는 조작자로는 조작레버(31), 정지키(32), 기록키(33), 메뉴키(34), 볼륨업키(35), 볼륨다운키(36) 및 홀드키(37)가 포함된다.

조작레버(31)는 적어도 상하방향으로 회전될 수 있는 조작자이다. 전형적으로, 조작레버(31)는 눌러질 수도 있다. 조작레버(31)를 조작함으로써, 음악데이터 등을 재생할 수 있는 조작, REW 및 AMS(Auto Music Search)조작, FF 및 AMS조작 또는 다른 조작을 실행할 수 있다. REW 및 AMS조작은 고속 되감기 및 선두-검색 조작인 반면에 FF 및 AMS조작은 고속-급지 및 선두-검색 조작이다.

정지키(32)는 음악데이터 등의 재생 또는 기록을 위한 조작이 정지되도록 요구하는 키이다.

기록키(33)는 음악데이터 등을 기록하기 위한 조작을 요구하는 키이다.

메뉴키(34)는 음악데이터 등의 편집이나 모드를 설정을 위해 조작되는 키이다. 편집모드에 있어서, 실제 편집조작은 조작레버(31)를 조작이나 이 메뉴키(34)를 이용하는 입력조작으로 실행될 수 있다.

볼륨업키(35) 및 볼륨다운키(36)는 각각 음악데이터를 재생하기 위해 조작 출력의 볼륨을 증가 또는 감소시키기 위한 키이다.

홀드키(37)는 다른 키의 조작기능을 유효 또는 무효하게 하는 키이다. 예를 들면, 드라이브장치(20)가 실행되는 동안에 키가 실수로 눌러지면, 오동작이 일어날 염려가 있다. 이 경우에, 홀드키(37)는 다른 키의 조작기능을 무효하게 한다.

물론 상기 기술된 키는 일예에 지나지 않는다. 더욱이, 드라이브장치(20)는 커서 이동키, 숫자키 및 조작다이얼 또는 조그다이얼과 같은 다른 키를 가질 수 있다.

전원을 온, 오프하는 키는 도시하지 않는다. 그러나, 조작레버(31)상에서 실행된 조작은 재생조작을 위한 요구가 전원을 온으로 하기 위한 조작으로서 해석될 수도 있다. 반면에, 조작이 정지키(32)상에 실행되기 때문에 소정 시간이 경과한 후 전원은 자동적으로 오프된다. 이러한 구조가 실행되면, 전원키는 불필요하게 된다. 물론, 전원키를 설치할 수도 있다.

설치된 조작자의 수, 조작자의 종류 및 조작자의 위치는 여러가지 방식으로 변경하는 것을 고려할 수 있다. 그러나, 필요한 최소의 것으로서 도 2a ~ 도 2e에 나타낸 조작자를 설치함으로써, 키수와 드라이브장치(20)의 크기 및 비용을 삭감할 수 있으며 조작성을 향상할 수 있다.

도 3은 드라이브장치(20)의 내부구성을 나타내는 블록도이다.

CPU(41)는 드라이브장치(20)의 중앙제어장치로서 사용하며 이후에 기술되는 것처럼 다른 소자를 제어한다.

CPU(41)에는 조작프로그램과 각종 정수를 저장한 내장된 ROM(41a) 및 작업영역으로서 사용되는 내장된 RAM(41b)이 포함된다.

조작부(30)는 상기 기술된 조작자(31 ~ 37)에 대응한다. CPU(41)는 조작부(30)로부터 수신된 조작 입력정보에 따라서 조작프로그램으로 규정된 제어조작을 실행한다.

플래시메모리(48)가 또한 설치된다. CPU(41)는 플래시메모리(48)에 시스템 설정정보 및 단자키 데이터를 저장시킬 수 있다. 시스템 설정정보는 음악기록모드, 재생볼륨 및 표시모드와 같이 각종 조작에 대한 정보이다. 단자키 데이터는 암호화 및 그 해독처리 또는 디코드 데이터의 처리에 사용된다.

리얼타임클럭(44)은 현재의 일시를 나타내는 소위 시계부로서 사용된다. CPU(41)는 리얼타임클럭(44)으로부터 날짜와 시간을 나타내는 데이터로부터 현재의 일시를 확인할 수 있다.

USB커넥터(43)는 USB커넥터(28)에 접속된 외부장치사이의 통신 인터페이스이다. 따라서, CPU(41)는 USB인터페이스(43)를 통해서 퍼스널 컴퓨터와 같은 외부장치와 데이터를교환할 수 있다. 외부장치에

교환된 데이터의 예는 제어데이터, 컴퓨터데이터, 화상데이터 및 오디오데이터이다.

레귤레이터(regulator)(46) 및 DC/DC컨버터(47)는 전원부로 사용한다. CPU(41)는 레귤레이터(46)에 전원 온요구를 내보냄으로써 전원을 온으로 한다. 그 요구를 수신하면, 레귤레이터(46)는 배터리로부터 전원 공급을 개시한다. A/D컨버터가 전원단자(29)에 접속되면, 레귤레이터(29)는 공급된 교류전압을 정류 및 평활하게 한다(smoothen).

레귤레이터(46)에 의해 발생된 전원전압은 DC/DC컨버터(47)에 의해 동작 전원전압(Vcc)으로서 각 블록에 공급되는 필요한 레벨의 전압으로 변환된다.

CPU(41)는 메모리 인터페이스(42)를 통해서 착탈기구(22)에 장착된 메모리시트(1)에 액세스할 수 있다. 따라서, CPU(41)는 데이터를 기록, 재생 및 편집을 실행하는 조작을 할 수 있다.

더욱이, CPU(41)는 표시드라이버(45)를 제어함으로써 표시부(21)에 필요한 화상을 표시하게 할 수도 있다. 화상 이외에, CPU(41)는 사용자를 위해 나타낸 조작 메뉴와 같은 정보 및 메모리시트(1)에 기록된 파일내용을 표시부(108)에 표시할 수 있다. CPU(41)는 메모리시트(1)로부터 독출된 화상데이터를 표시할 수 있다. 메모리시트(1)에 저장된 화상데이터는 전형적으로 동화상 또는 정지화상을 나타낸다.

상기 기술한 것처럼, 실시예에는 오디오신호를 입출력하기 위해서 디지털 입력단자(27), 마이크로폰 입력단자(25), 라인입력단자(26), 헤드폰 입력단자(23) 및 라인출력단자(24)가 설치된다.

이들 단자를 통하여 신호 입출력을 처리하는 오디오신호 처리시스템은 암호화/복호화로서 사용되는 SAM(Security Application Module)(50), DSP(Digital Signal Processor)(49) 및 A/D(아날로그에서 디지털로) 및 D/A(디지털에서 아날로그로)변환부(54)(이하에서는 ADDA변환부라고 칭한다), 전원증폭기(56), 마이크로폰 증폭기(53), 광입력모듈(optical input module)(51) 및 디지털 입력모듈(52)으로 구성된다.

SAM(50)은 CPU(41)와 DSP(49)사이에서 교환된 데이터를 암호화 및 복호화하고, CPU(41)와 암호키로서 사용되는 단자키 및 식별자(identifier)를 교환한다. 즉, SAM(50)은 단자키를 이용하여 데이터를 암호화 및 복호화 한다.

SAM(50)에 의해 암호화 및 암호해제된 데이터에는 음악데이터와 같이 데이터파일에 저장된 실제 데이터 이외에 관리정보 및 부가정보가 포함되는 것에 주목해야 한다. 이후에 기술되는 것처럼, 관리정보는 트랙관리 정보파일에 저장된 데이터이며, 부가정보는 부가정보파일에 저장된 데이터이다.

DSP(49)는 CPU(41)로 발생된 지시에 따라서 오디오데이터를 압축 및 압축해제한다.

디지털 입력단자(52)는 광입력모듈(51)에 의해 입력된 디지털 오디오데이터의 입력 인터페이스처리를 실행한다.

ADDA변환부(54)는 오디오신호의 A/D변환 및 D/A변환을 실행한다.

이들 블록은 다음과 같이 오디오신호를 입력하고 출력한다.

디지털 오디오데이터로서 외부기로부터 광케이블을 통해서 디지털 입력단자(27)에 공급된 신호는 전송 포맷에 따라서 디지털 입력부(52)에 의해 수신처리가 실행되기 전에 광입력모듈(51)에 의해 광전변환처리된다. 수신처리에서 추출된 디지털 오디오데이터는 DSP(49)에 의해 압축되며 CPU(41)에 공급되어 메모리시트(1)에 기록된다. 물론, DSP(49)에 의해 출력된 데이터는 어떤 경우에 SAM(50)에 의해 암호화될 수 있다.

마이크로폰 입력단자(25)에 접속된 마이크로폰으로 공급된 입력 오디오신호는 ADDA변환부(54)에서 A/D변환되기 전에 마이크로폰 증폭기(53)에 의해 증폭된다. A/D변환처리의 결과는 디지털 오디오데이터로서 DSP(49)에 공급된다. DSP(49)에서의 압축처리 그리고 어떤 경우에, SAM(50)에서 복호처리를 실행한 후, 데이터는 CPU(41)에 공급되어 메모리시트(1)에 기록된다.

한편, 예를 들면 메모리시트(1)로부터 독출된 오디오데이터를 출력하기 위한 조작에 있어서 CPU(41)는 오디오데이터에 대해서 압축해제처리를 실행하도록 DSP(49)에 요구하고, 어떤 경우에 데이터에 대해서 복호처리를 실행하도록 SAM(50)에 요구한다. 이들 처리를 종료하는 디지털 오디오데이터는 ADDA변환부(54)에 의해 전원증폭기(56)로 변환된다.

전원증폭기(56)는 신호가 헤드폰으로 출력되도록 하는 증폭처리와 라인출력으로서 출력되도록 하는 증폭처리를 실행하고, 증폭된 신호를 입력단자(23)와 라인출력단자(24)에 출력한다.

이후에 기술되는 것처럼, 드라이브장치(20)는 메모리시트(1)로부터 독출된 압축된 오디오데이터와 디지털 입력단자(27), 마이크로폰 입력단자(25) 또는 라인입력단자(26)로 입력되고, DSP(49)로 압축된 오디오데이터에 대해서 SAM(50)에서 암호화처리를 실행하고, 압축된 데이터를 USB커넥터(43) 및 USB단자(28)를 거쳐서 퍼스널 컴퓨터와 같은 외부장치에 공급할 수 있다.

더욱이, USB단자(28)에 접속된 외부장치로부터 입력된 오디오데이터는 USB단자(28)에 접속된 외부장치로 복귀되기 전에 SAM(50)에 있어서 암호화처리될 수 있다.

반면에, 드라이브장치(20)는 외부장치로부터 수신된 오디오데이터를 CPU(41)에 의해 USB인터페이스(53)를 거쳐서 메모리시트(1)에 기록하거나 필요에 따라서, SAM(50)에서 복호화하고, 헤드폰단자(23) 또는 라인출력단자(24)에 데이터를 출력하기 전에 DSP(49)에서 압축해제처리를 실행한다. 더욱이, 드라이브장치(20)는 USB인터페이스(43)를 통해서 퍼스널 컴퓨터와 같은 외부장치로 데이터를 전송할 수 있다. 이러한 외부장치로 전송된 데이터의 예는 암호해제처리를 종료하는 수신된 데이터이다.

도 3에 나타난 드라이브장치(20)의 구성은 단지 일예임에 주목해야 한다. 즉, 드라이브장치(20)는 이러한 구성에 한정되지 않는다.



예를 들면, 구성은 오디오데이터를 출력하기 위한 내장된 스피커를 포함하는 것을 고려할 수 있다. 이 경우에, 전원증폭기(56)에 의해 출력된 신호는 스피커에 공급되고, 그 신호에 의거해 음성을 출력한다.

### 3. 전형적인 시스템 접속

도 4는 드라이브장치(20)를 중심성분으로서 사용하는 전형적인 시스템접속을 나타내는 설명도이다.

드라이브장치(20)는 독립형 소자로서 사용될 수 있고, 또한 드라이브장치(20)와 통신하는 퍼스널 컴퓨터(11)와 같은 다른 장치와 접속하는 시스템에서 사용될 수 있다.

상기 기술한 바와 같이, 메모리시트(1)를 드라이브장치(20)에 장착함으로써, 드라이브장치(20)는 메모리시트(1)에 데이터를 기록하고 메모리시트(1)로부터 재생할 수 있는 독립형 소자로서 기능할 수 있다.

예를 들면, 드라이브장치(20)에 장착된 음악데이터를 기록하는 메모리시트(1)를 갖추는 경우에, 도 4에 나타난 것처럼 사용자는 헤드폰(12)을 접속함으로써 재생된 음악을 즐길 수 있다.

상기 기술한 바와 같이, CD플레이어(10)와 같은 외부재생장치는 케이블(13)에 의해 라인출력단자(26) 또는 디지털 입력단자(27)에 접속될 수 있다. 드라이브장치(20)에 접속된 CD플레이어(10)로, 드라이브장치(20)는 CD플레이어(10)로부터 재생오디오신호를 수신하고 그 신호를 메모리시트(1)로 기록할 수 있다.

더욱이, 드라이브장치(20)는 여기에 접속된 마이크로폰에 의해 잡음된 음성을 메모리시트(1)에 기록할 수도 있다. 드라이브장치(20)는 여기에 접속된 MD레코더와 같은 기록장치에 데이터를 공급하여 기록 장치에 장착된 기록매체에 기록하도록 할 수 있다. 마이크로폰 및 기록장치 자체가 도면에 도시되지 않은 것에 주목해야 한다.

퍼스널 컴퓨터(11)와 같은 정보장치는 USB(Universal Serial Bus)라이벌(14)에 의해 드라이브장치(20)에 접속될 수 있다. 이 경우에, 드라이브장치(20)는 퍼스널 컴퓨터(11)로부터 수신된 데이터를 메모리시트(1)에 기록하거나 메모리시트(1)로부터 재생된 데이터를 복사 또는 이동(move)조작으로 퍼스널 컴퓨터(11)로 전송할 수 있다.

이러한 복사 또는 이동의 목적지는 퍼스널 컴퓨터(11)에서 사용되는 HDD(11a)이다.

또한, 도면은 퍼스널 컴퓨터(11)에 설치된 스피커(11b) 및 CD-ROM드라이브장치(11c)를 나타내는 것에 주목해야 한다. 이 경우에, 드라이브장치(20)는 퍼스널 컴퓨터(11)와 접속하여 조작을 실행할 수 있다. 예를 들면, 드라이브장치(20)는 CD-ROM드라이브(11c)로부터 수신된 오디오데이터를 메모리시트(1)에 기록하거나 CD-ROM드라이브(11c)로부터 수신된 암호화된 오디오데이터를 복호하고, 그 결과를 퍼스널 컴퓨터(11)에 재전송할 수 있다.

더욱이, 드라이브장치(20)는 오디오데이터를 퍼스널 컴퓨터(11)에 전송하여 스피커(11b)로부터 음성으로서 출력할 수 있다.

상기 기술한 바와 같이, 각종 기기가 드라이브장치(20)에 접속될 수 있다. 결과적으로, 드라이브장치(20)가 사용자에게 의해 실행되는 동안, 기록 및 재생조작을 실행할 수 있다. 더욱이, 드라이브장치(20)를 가정이나 직장에 놓인 각종 기기에 접속함으로써, 드라이브장치(20)는 시스템조작을 실행할 수 있다.

더욱이, 본 실시예에 의해 설치된 드라이브장치(20)는 표시부를 갖기 때문에, 드라이브장치(20)는 메모리시트(1)에 기록된 문자데이터 및 화상데이터와 같은 정보를 재생하기 위해서 독립형 소자로서 기능할 수 있고 이것들을 표시부에 표시할 수 있다. 물론, 메모리시트(1)로부터 재생된 문자데이터 및 화상 데이터와 같은 정보는 퍼스널 컴퓨터(11)의 모니터화면에 표시될 수 있다.

### 4. 파일시스템

#### 4-1 처리구조 및 데이터구조

다음에, 메모리시트(1)를 사용하는 본 실시예의 파일시스템을 설명한다.

도 5는 메모리시트를 기록매체로서 사용하는 컴퓨터시스템의 파일시스템 처리계통을 설명한다.

도면에 도시된 바와 같이, 파일시스템 처리계통에 있어서 응용처리계통은 상위계통이며 파일시스템 처리계통, 논리어드레스(logical address) 관리계통, 물리어드레스(physical address) 관리계통 및 플래시메모리 액세스계통이 이후에 순서대로 계통을 이룬다. 이 계통구조에 있어서, 파일관리 처리계통은 FAT(File Allocation Table)파일시스템이다. 물리어드레스는 플래시메모리의 각 블록에 할당된다. 따라서, 물리어드레스와 블록간의 관계는 고정된다. 논리어드레스는 파일관리 처리계통에서 논리적으로 취급된 어드레스이다.

도 6a ~ 도 6f는 메모리시트(1)에 사용된 플래시메모리에 저장된 데이터의 물리적구조를 나타낸다.

도 6a에 나타난 바와 같이, 메모리시트(1)에 사용된 플래시메모리내의 세그먼트(segment)로 명명된 데이터부는 고정길이를 가지는 소정의 블록수로 분할되고, 각 블록은 고정길이를 가지는 소정수의 페이지로 분할된다. 데이터는 메모리시트(1)로부터 블록단위로 삭제되지만 기입과 독출은 페이지단위로 행해진다. 블록과 페이지는 동일한 크기를 갖는다. 1블록은 페이지 0 ~ 페이지 m을 포함한다.

1블록은 8KB(kilobytes) 또는 16KB의 크기를 갖는 반면에, 1페이지는 512b의 크기를 갖는다. 8KB 블록크기의 경우에 있어서, 메모리시트(1)의 총 저장용량은 512블록에 대응하는 4MB가 되거나 1,024블록에 대응하는 8MB가 될 수 있다. 반면에, 16KB 블록크기인 경우에 메모리시트(1)의 총 저장용량은 1,024블록에 대응하는 16MB, 2,048블록에 대응하는 32MB 또는 4,096블록에 대응하는 64MB가 될 수 있다.

도 6c에 나타낸 바와 같이, 1페이지는 512bytes의 크기를 갖는 데이터 부분과 16bytes의 크기를 갖는 용장부분을 포함한다. 용장부분의 구조는 도 6d에 나타난다. 우선, 3bytes의 크기를 갖는 재기입(overwrite)부분은 용장부분의 선두로서 사용된다. 재기입 부분은 갱신 데이터에 대한 조작에 따라서 재기입되는 영역이다. 3bytes의 재기입 부분은 선두로부터 각각 순서대로 배치된 블록상태, 페이지상태 및 갱신상태이다.

마지막으로, 용장부분(redundancy portion)의 나머지 13bytes의 데이터는 데이터부분의 내용에 따라서 고정된다. 13bytes는 1bytes의 관리플래그(management flags), 2bytes의 논리어드레스, 5bytes의 포맷유지(format reserve)영역, 2bytes의 분산정보 ECC(Error Correction Code) 및 3bytes의 데이터 ECC이다. 분산정보 ECC는 관리플래그, 논리어드레스 및 포맷유지영역에서 오차를 보정하기 위한 용장데이터이다. 반면에, 데이터 ECC는 도 6c에 나타낸 512bytes 데이터부분에서 오차를 보정하기 위한 용장데이터이다.

관리플래그는 시스템플래그, 변환 테이블플래그, 복사금지 지정플래그 및 액세스 허가플래그이다. 시스템 플래그 1은 사용자 블록을 나타내고, 시스템 플래그 0은 부트블록(boot block)을 나타낸다. 변환 테이블플래그 1은 무효를 나타내고, 변환테이블 0은 테이블블록을 나타낸다. 복사금지 지정플래그 1은 복사조작이 허가(OK)된 것을 나타내고, 복사금지 지정플래그 0은 복사조작이 허가되지 않은 것(NB)을 의미한다. 액세스 허가플래그 0은 독출 보호를 의미한다.

도 6a에 나타낸 바와 같이, 세그먼트의 선두에서 2개의 블록 즉, 블록 0과 1은 부트 블록인 반면에 나머지는 사용자 블록으로서 정보블록이다.

백업블록으로 사용된, 도 6b에 나타낸 부트블록의 블록 1은 블록 0으로 기입된 동일한 1개 데이터를 갖는다. 부트블록은 메모리시트(1)의 카드내에 유효한 블록의 선두블록이다. 부트블록은 메모리시트(1)가 드라이브장치(20)에 장착될 때 초기에 액세스되는 블록이다.

도 6e에 나타낸 바와 같이, 부트블록의 선두블록에는 블록의 선두의 페이지 0에는 선두(header), 시스템 목록 및 부트 및 분배정보가 포함된다. 페이지 1에서, 사용 금지된 블록내에 저장되는 반면에, 페이지 2에서는, CIS(Card Information Structure) 및 IDI(Identify Drive Information)가 저장된다.

제 1부트블록의 선두는 부트블록 ID 및 부트블록내의 유효목록 번호를 기록하는데 사용된다. 시스템 목록은 사용금지된 블록의 개시위치, 데이터의 크기 및 데이터의 유형을 기록하는데 사용된다. 시스템목록은 CIS 및 IDI데이터의 개시위치, 데이터의 크기 및 데이터의 유형을 기록하는데 사용된다. 부트 및 분배정보에는 메모리시트(1)의 유형, 블록크기, 블록수, 총 블록수, 안정성이 지지되는지의 여부를 나타내는 정보와 제조일자를 포함하는 메모리시트(1)의 제조와 관련된 데이터를 기록하는데 사용된다. 메모리시트(1)의 유형은 메모리시트(1)가 독출만 가능한 메모리, 읽고 쓸수 있는 메모리 또는 하이브리드 메모리, 즉 2종류의 메모리 금지 특성인지의 여부를 나타낸다.

상기 기술은 메모리시트(1)에 저장된 데이터의 구조를 설명한다.

그러나, 플래시메모리의 절연막은 메모리에 데이터를 기입하는 조작으로 인해 열화될 수 있다. 이러한 문제를 해결하기 위해서, 기입조작 횟수가 제한된다. 따라서, 동일한 저장영역 또는 동일한 블록에 반복적으로 그리고 지속적으로 액세스가 이루어지는 것을 방지할 필요가 있다. 이러한 액세스는 다음과 같은 파일시스템에 의해 물리어드레스에 저장된 데이터를 갱신하기 위해서 조작중에 회피될 수 있다. 물리어드레스는 논리어드레스로서 메모리시트(1)의 파일시스템에 의해 보여진다. 따라서, 어플리케이션에 의해 이루어지는 논리어드레스의 액세스는 파일시스템에 의해 임의로 결정된 물리어드레스에 대해서 액세스로서 파일시스템에 의해 실행될 수 있다. 결과적으로, 파일시스템은 데이터 갱신조작되는 블록의 데이터를 갱신하고, 데이터 갱신조작되는 블록에 데이터를 재기입하는 대신에 다른 미사용된 블록에 갱신된 데이터를 재기입할 수 있다. 그러므로, 데이터 갱신조작 이후에 물리어드레스와 논리어드레스사이의 관계는 데이터 갱신조작 이전의 물리어드레스와 논리어드레스 사이의 관계와 다르다. 데이터 갱신조작되는 블록에 데이터를 재기입하는 대신에 갱신된 데이터를 미사용된 블록에 재기입하는 조작은 스왑(swap)처리로 칭한다. 상기 기술한 바와 같은 스왑처리를 실행함으로써, 동일한 블록에 반복적으로 그리고 지속적으로 액세스가 이루어지는 것을 방지한다. 결과적으로, 메모리시트(1)로서 사용된 플래시메모리의 수명은 연장될 수 있다.

논리어드레스는 데이터가 블록에 기입될 때 데이터에 할당된다. 따라서, 데이터 갱신조작 이후의 블록이 상기 기술한 바와 같이 데이터 갱신조작 이전의 블록과 다르더라도, FAT에서 블록에 저장된 데이터에 대한 논리어드레스의 할당이 변경되지 않으며, 따라서 이후에 논리어드레스는 데이터에 정확하게 액세스할 수 있다. 스왑처리가 논리어드레스와 물리어드레스 사이의 관계를 변화시키기 때문에, 양 어드레스 사이의 관계를 나타내는 논리어드레스/물리어드레스 변환테이블이 필요하다. 이 논리어드레스/물리어드레스 변환테이블을 참조하여, 어플리케이션에 의해 지정된 논리어드레스에 대응하는 물리어드레스는 테이블을 제어하는 FAT시스템에 의해 확인될 수 있다. 결과적으로, 확인된 물리어드레스에서 블록에 대한 액세스가 이루어질 수 있다.

논리어드레스/물리어드레스 변환테이블 드라이브장치(20)내에 내장된 RAM(41 b)에 저장되지만 메모리시트(1)에도 저장될 수 있다.

대략, 2bytes의 길이를 가지는 논리어드레스는 오름차순으로 논리어드레스/물리어드레스 변환테이블에 배치되고 2bytes의 길이를 가지는 물리어드레스는 각각의 논리어드레스에 할당된다. 플래시메모리의 최대용량은 128MB(또는 8,192블록)이고, 물리어드레스가 각각의 블록에 할당되기 때문에, 8,192어드레스이면 충분하다. 논리어드레스/물리어드레스 변환테이블은 세그먼트마다 제어되고, 메모리시트(1)의 용량에 따라서 그 크기는 증가한다. 메모리시트(1)의 저장용량은 8MB 또는 2세그먼트로 가정하고, 2페이지는 2세그먼트마다 논리어드레스/물리어드레스 변환테이블에 할당된다.

논리어드레스/물리어드레스 변환테이블은 상기 기술한 바와 같이 메모리시트(1)에 저장될 수도 있다. 이 경우에, 각 페이지의 용장부분에 있어서 관리플래그의 소정 1비트는 페이지에 대한 논리어드레스/물리어드레스 변환테이블이 메모리시트(1)에 저장되는 것을 나타내도록 설정된다.

본 실시예에 의해 설치된 메모리시트(1)는 디스크형 기록매체와 같이 동일한 방식으로 퍼스널 컴퓨터의 FAT파일시스템에 의해 사용될 수 있다. 메모리시트(1)에는 IPL(Initial Program Loader)영역, FAT영역 및 도 6a에 나타내지 않은 루트 디렉토리영역이 포함된다. IPL영역은 초기에 드라이브장치(20)의 메모리에 장착되는 프로그램의 어드레스를 기록하고 각종 정보를 메모리에 기록하는데 사용된다. FAT영역은 블록 또는 클러스터와 관련된 사항을 기록하는데 사용된다. 구체적으로, FAT영역은 미사용된 블록, 다음 블록 수, 불량 블록 및 최후 블록을 나타내는 값을 규정하는데 사용된다. 루트디렉토리 영역은 파일속성, 경신일시 및 개시 클러스터 및 파일크기를 포함하는 디렉토리 목록을 저장하는데 사용된다.

본 실시의 형태에서는 상술한 메모리시트(1)의 포맷으로 규정되는 파일관리시스템과는 별개로 음악용 파일에 대하여 파일관리정보, 즉 구체적으로 설명하면 후술하는 트랙정보 관리파일을 규정하고 있다. 트랙정보 관리파일은 메모리시트(1)의 사용자블록을 이용하여 기록된다. 기록된 트랙정보 관리파일에 의해, 메모리시트(1)상의 FAT가 파괴되어도, 파일의 복구를 가능하게 할 수 있다.

트랙정보 관리파일은 CPU(41)에 의해 작성된다. 예를 들면, 최초로 전원을 온으로 할 때에, 메모리시트(1)가 장착되어 있는지의 여부를 판정한다. 메모리시트(1)가 장착되어 있을 때, 그 메모리시트(1)는 인증된다. 인증에 의해 메모리시트(1)가 정규의 메모리시트인 것이 확인되면, CPU(41)는 메모리시트(1)내의 부트블록(boot block)으로부터 정보를 독출한다. 그리고, CPU(41)는 논리어드레스/물리어드레스 변환테이블을 독출한다.

CPU(41)에 의해 독출된 데이터는 RAM(41b)에 격납된다. 사용자가 구입하여 일반적으로 사용하는 메모리시트(1)의 경우에도, 출하시에는 FAT나 루트디렉토리(root directory)가 메모리시트(1)에 저장되어 있다.

트랙정보 관리파일은 각 기록조작에 따라서 작성되어 갱신된다.

즉, 결판적으로 저장자와 조작 등에 기초하여 오디오데이터를 후술하는 데이터파일로서 기록한 경우, 그 기록조작후에 FAT 및 트랙정보 관리파일이 갱신된다. 파일을 갱신하는 조작에 있어서, 구체적으로는 오디오데이터를 기록하기 위한 조작을 완료할 때이다. RAM(41b)상에 저장된 FAT 및 트랙정보 관리파일이 갱신된다. 그리고, 메모리시트(1)를 때어낼 때 또는 전원을 오프할 때에, RAM(41b)으로부터 메모리시트(1)에 최종적인 FAT 및 트랙정보 관리파일이 이동된다.

또한, 오디오데이터를 기록하기 위한 조작이 완료할 때마다, 메모리시트(1)상에 저장된 FAT 및 트랙정보 관리파일을 갱신할 수 있다는 것에 주목해야 한다. 편집을 행한 경우에도, 트랙정보 관리파일의 데이터가 갱신된다.

또, 본 실시의 형태에서는 부가정보를 설명하는 부가정보파일도 작성, 갱신되고, 메모리시트(1)에 기록된다. 부가정보파일의 작성, 갱신은 트랙정보 관리파일과 같은 방식으로 된다.

부가정보는 외부기로부터의 송신이나 사용자의 입력조작에 의해 드라이브장치(20)에 공급된다. 그 공급된 부가정보는 CPU(41)에 의해 메모리시트(41)에 기록된다. 부가정보의 기록동작에 대한 상세한 설명은 다른 동작과 관련하여 후술한다.

#### 4-2. 디렉토리 구성

다음으로, 메모리시트(1)에 기록되는 파일구조에 대해서 설명한다.

우선, 일반적인 디렉토리의 구성에 대하여 도 7에 나타낸다.

상술한 바와 같이, 메모리시트(1)를 사용하여 관리할 수 있는 메인데이터로서는 동화상 데이터, 정지화상 데이터, 음성데이터(voice data), 오디오 데이터, 음악데이터, 제어용 데이터 등이 있다. 또한, 디렉토리구조에 있어서, 루트디렉토리(root directory)는 VOICE, DCIM, M0xxxxnn, AVCTL, HIFI의 서브디렉토리를 갖는다. VOICE는 보이스용 디렉토리이고, DCIM은 정지화상용 디렉토리이다. AVCTL은 제어용 디렉토리이고, HIFI는 음악용 디렉토리이다.

이 실시의 형태에서는 음악이나 오디오 데이터를 기록 및 재생하는 경우를 예로 설명하기 때문에, 이하에서는 HIFI 음악용 디렉토리에 대해서만 다음과 같이 설명한다.

HIFI 음악용 디렉토리는 트랙정보 관리파일(TRKLIST.MSF)과, 트랙정보 관리파일(TRKLIST.MSF)의 백업파일(TRKLISTB.MSF)과, 부가정보파일(INFLIST.MSF)과, 데이터파일(A30nnnnn.MSA)을 갖는다.

데이터파일(A30nnnnn.MSA)은 실제의 음악데이터를 기록하는 파일, 즉 ATRAC3(Adaptive Transform Acoustic Coding)방식에 따라서 압축된 오디오데이터이다. 1개의 데이터파일(A30nnnnn.MSA)이 1개의 악곡을 저장하기 위해 사용된다. 또한, 이하의 설명에서는 데이터파일을 트랙이라고 칭하는 것에 주목해야 한다.

트랙정보 관리파일(TRKLIST.MSF)은 음악용 디렉토리내의 관리파일이며, CD시스템이나 MD시스템에 있어서의 TOC와 마찬가지로 데이터파일에 저장되는 각 악곡을 관리하는 파일이다.

또, 이 트랙정보 관리파일(TRKLIST.MSF)은 NAME1 및 NAME2를 포함한다. NAME1은 메모리시트(1) 전체의 명칭, 곡명을 1바이트 코드로 기술하는 블록이다. 즉, NAME1은 ASCII/8859-1의 문자코드에 의해 곡명과 같은 데이터가 기술된다. 한편, NAME2는 메모리시트(1) 전체의 명칭이나 곡명 등을 2바이트 코드로 기술하는 블록이다. 즉, MS-JIS 문자코드, 한국어 또는 중국어에 의해 곡명데이터 등이 기술된다.

부가정보파일(INFLIST.MSF)은 메모리시트(1)의 전체 혹은 예를 들면 악곡을 저장하기 위해 사용되는 각 데이터파일에 대응하는 부가정보를 관리 및 기록하는 파일이다. 구체적으로는 아티스트명, ISRC코드, 타임스탬프(time stamp), 정지화상 데이터 등의 부가정보파일(INFLIST.MSF)에 저장되는 부가정보를

포함한다.

#### 4-3. 관리구조(Management Structure)

도 8a~도 8c는 음악용 디렉토리에 포함되는 트랙정보 관리파일(TRKLIST.MSF)의 NAME1 및 NAME2, 데이터파일(A3Dnnnn.MSA), 부가정보파일(INFLIST.MSF)사이의 관계를 나타내는 도면이다.

또한, 본 명세서에서의 설명 또는 도면상에서, 예를 들면 0x0010 등의 0x를 부가한 수식은 16진 표기인 것을 의미한다.

트랙정보 관리파일(TRKLIST.MSF)은 전체 64K바이트(= 16K× 4)의 고정길이를 갖는다. 트랙정보 관리파일(TRKLIST.MSF)의 32K바이트 부분은 트랙이나 데이터파일을 관리하는 파라미터를 저장하기 위하여 사용된다. 나머지 32K바이트 부분은 상술된 NAME1 및 NAME2에 할당된다. 곡명 등의 정보를 기술한 NAME1 및 NAME2는 트랙정보 관리파일과 별도로 취급하는 것도 실현할 수 있다. 그러나, RAM용량이 작은 시스템의 경우, 분리된 파일대신에 1개의 관리파일로서 트랙정보 관리파일(TRKLIST.MSF)과 곡명파일을 관리하여 조작이 더 쉽게 될 수 있다.

트랙정보 관리파일(TRKLIST.MSF)에서는 파일내의 트랙정보영역(TRKINF-nnnn) 및 파트정보영역(PRTINF-nnnn)에 의해 데이터파일(A3Dnnnn.MSA) 및 부가정보파일(INFLIST.MSF)이 관리된다.

예를 들면, 1개의 데이터파일(A3D(n))에 대해서, 1개의 트랙정보영역(TRKINF-(n)) 및 파트정보영역(PRTINF-(n))에, 포인터 및 다양한 종류의 다른 정보가 기술된다. 도 8a~도 8c에 나타난 예에서는 4개의 파트(P1~P4)로 구성된 데이터파일(A3D(n))을 관리하기 위하여, 트랙정보영역(TRKINF-(n)) 및 파트정보영역(PRTINF-(n))을 사용한다.

이 경우, 그 데이터파일(A3D(n))에 대응하는 곡명 등의 문자정보가 NAME1(n), NAME2(n)에 기술된다.

또, 상술되는 바와 같이, 트랙정보영역(TRKINF-(n))은 부가정보파일(INFLIST.MSF)내의 인덱스포인터(INF-(n))를 나타내는 포인터(pointer)가 기술되어 있다. 더욱이, 부가정보파일(INFLIST.MSF)내의 인덱스포인터(INF-(n))는 실제의 부가정보내용이 기술된 부가정보유닛(INFU-(n))을 나타낸다.

상술된 방식에 따라서 데이터파일에 대응하는 부가정보의 유닛(unit)이 관리된다.

또한, 부가정보파일(INFLIST.MSF)내의 인덱스포인터(INF-(n))는 본 발명에서 제 1포인터로서 칭한다는 것에 주목해야 한다. 한편, 트랙정보영역(TRKINF-(n))내의 상기 인덱스포인터를 나타내는 포인터는 본 발명에서 제 2포인터로서 칭한다.

계속하여, 도 9a~도 9c를 참조하여 트랙으로서의 악곡과 데이터파일사이의 관계에 대해서 설명한다.

1트랙은 1곡을 의미한다. 메모리시트(1)에 기록할 수 있는 트랙의 수는 일반적으로 최대 400트랙으로 제한된다. 1곡은 1개의 데이터파일로 저장된다. 데이터파일은 ATRAC3방식에 의해 압축된 오디오데이터이다. 또, 데이터파일은 클러스터(cluster)라 불리는 단위로 메모리시트(1)에 기록된다. 1클러스터 유닛의 일반적인 사이즈는 16KB의 용량이다. 따라서, 1클러스터에는 복수의 데이터파일이 혼합되지 못한다.

메모리시트(1)내의 데이터를 소거할 때의 최소단위가 1블록이다. 음악데이터를 기록하기 위하여 사용하는 메모리시트(1)의 경우, 블록은 클러스터와 같은 의미를 가지며, 또 1클러스터는 1섹터(sector)로 정의되어 있다.

1곡은 기본적으로 1파트로 구성된다. 그러나, 편집의 결과로서 얻어지는 악곡은 복수의 파트를 구성한다. 곡내의 파트사이의 연결(link)은 트랙정보 관리파일(TRKLIST.MSF)에 있어서의 파트정보영역(PRTINF-nnnn)에 의해 기술된다.

파트(part)는 기록동작의 개시에서 시작하고 기록동작의 마지막의 정지까지의 연속한 시간내에서 기록된 데이터의 단위를 의미한다. 통상적으로, 1트랙이 1파트로 구성된다. 파트의 최대값에 제한이 있다. 파트의 수를 P라 하고, 트랙의 수를 T라 하고, T는 1~400의 범위의 값을 가지면, 파트의 수와 트랙의 수사이에는  $P=2043-4 \times T$ 의 관계가 있다. 1트랙이 2,039파트로 구성된다고 가정하면 2곡목으로 할당될 수 있는 충분한 파트는 없게 되며, 2곡목의 데이터파일을 작성할 수 없게 된다.

파트의 최소단위는 SU이라 약기된 사운드유닛이다. SU는 ATRAC3방식에 따라서 압축된 오디오데이터의 최소의 데이터단위이다. 즉, SU는 몇 백바이트의 사이즈를 갖는 데이터단위이다. 44.1kHz의 주파수에서 샘플링동작의 결과로 얻어진 1024샘플(즉,  $1024 \times 16\text{비트} \times 2\text{채널}$ )의 오디오데이터를 약 1/10로 압축한 결과의 데이터 유닛이 SU이다. 1SU는 시간으로 환산하여 약 23m초가 된다. 통상적으로, 수천 개의 SU에 의해 1개의 파트가 구성된다.

도 9a~도 9c는 CD 등의 기록매체로부터의 오디오데이터를 2곡 연속하여 기록한 경우의 파일구성을 나타내는 도면이다.

더 상세하게 설명하면, 도 9a는 5클러스터로 구성되는 제 1곡목에 대한 데이터파일(#1)의 구성을 나타내는 도면이고, 도 9c는 6클러스터로 구성되는 제 2곡목에 대한 데이터파일(#2)의 구성을 나타내는 도면이다.

제 1곡목에 대한 데이터파일의 마지막과 제 2곡목에 대한 데이터파일의 시작에서 1클러스터에 2개의 파일이 존재하는 것이 허가되지 않기 때문에, 데이터파일(#1)의 마지막 클러스터에 계속하여 다음의 클러스터의 최초에서 데이터파일(#2)이 작성된다. 따라서, 데이터파일(#1) 즉 제 1곡목이 클러스터의 도중에서 종료될 경우, 그 클러스터의 나머지 부분에는 도 9b에 나타난 데이터나 SU가 존재하지 않게 된다.

제 2곡목의 데이터파일(#2)에서도 상술한 바와 마찬가지로 된다.

상술한 예에 있어서, 데이터파일(#1, #2)은 각각 1파트로 구성된다.

메모리시트(1)에 저장된 데이터파일을 편집하기 위한 작업으로서는 디바이드(divide), 결합(combine), 삭제(erase), 이동(move)의 4종류의 처리가 규정된다.

디바이드는 1개의 곡을 2개로 분할하는 것이다. 따라서, 디바이드처리가 실행되면, 총 곡수가 1개 증가한다. 디바이드처리에 있어서, 1개의 파일을 파일시스템상에서 분할하여 2개의 파일로 하고, 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다. 삭제는 곡의 삭제를 실행하는 것이다. 소거된 이하의 곡 번호가 1감소한다. 이동처리는 곡의 순번을 변경하는 것이다. 또한, 이동처리는 메모리시트(1)로부터의 곡을 하드디스크 등의 다른 매체로 이동시킬 수 있다. 복사(copy)처리는 오리지널곡의 복제를 작성하는 조작의 것에 대하여 이동처리는 단순히 곡의 위치를 변경하기 위한 처리이다. 따라서, 이동처리에 의해 곡의 복제가 발생하지는 않는다.

도 10은 도 9a~도 9c에 나타난 2개의 곡, 즉 데이터파일(#1, #2)을 결합한 처리결과를 나타내는 도면이다. 결합처리를 실행함으로써, 데이터파일(#1, #2)은 2개의 파트로 이루어진 1개의 새로운 데이터파일(#1)로 결합된다.

또, 도 11a~도 11b는 도 9a에 나타난 1개의 곡, 즉 데이터파일(#1)을 클러스터(2)의 도중에서 디바이드한 처리결과를 나타내는 도면이다.

데이터파일(#1)에 대한 디바이드처리에 의해 새로운 데이터파일(#1, #2)이 발생한다. 새로운 데이터파일(#1)은 클러스터(0, 1) 및 클러스터(2)의 전속으로 이루어지며, 새로운 데이터파일(#2)은 클러스터(2)의 후속과 클러스터(3, 4)로 이루어진다.

상술한 각종의 편집처리로 데이터파일에 저장된 데이터자료를 갱신하는 경우, 시간이 걸리는 동시에 처리부담이 크게 된다. 이러한 문제점을 해결하기 위하여, 편집중(edit point)을 포함하는 블록 또는 클러스터의 트랙정보 관리파일(TRKLIST.MSF)만이 갱신된다. 이 때문에, 파트(part)라는 개념이 도입되고 있다.

#### 4-4. 트랙정보 관리파일 TRKLIST.MSF

트랙정보 관리파일(TRKLIST.MSF)의 자세한 구성은 다음과 같이 도 12a, 도 12b, 도 13, 도 14, 도 15, 도 16을 참조하여 설명된다.

도 8a~도 8c에 도시된 바와 같이, 트랙정보 관리파일(TRKLIST.MSF)은 각각 16KB의 크기를 띠며 64KB가 되는 4개의 클러스터 또는 4개의 블록들로 구성된다. 그러나, 도 12a는 제1과 제2 클러스터들로만 구성된 32KB의 제1 반쪽만을 도시하고 있다.

이러한 32KB의 크기를 가지는 제1 반쪽 영역은 트랙들 또는 데이터 파일들을 제어하는데 이용된다.

도 7를 참조하여 이미 설명된 바와 같이, 트랙정보 관리파일(TRKLIST.MSF) 뿐만 아니라, 트랙정보 관리파일(TRKLIST.MSF)의 복사파일인 TRKLISTB.MSF가 생성된다는 것을 주목해야 한다. 복사파일은 원래 파일과 동일한 구조를 가지고 있으므로, 복사 파일의 설명은 트랙정보 관리파일(TRKLIST.MSF)의 다음 설명으로부터 생략된다.

더욱이, 각 파일의 다음 설명에서 이용되는 용어 슬롯(slot)은 파일의 시작점으로부터 시작되는 8 또는 16바이트들의 한정 단위로 정의된다. 트랙정보 관리파일(TRKLIST.MSF)의 경우에는, 슬롯의 크기가 16바이트들이다.

도 12a, 도 12b, 도 17, 도 19에 도시된 도면들에서는, 수평 방향내의 열(row)은 8 또는 16바이트들로 구성되어 있다. 그러므로, 이러한 열은 한 개의 슬롯에 대응한다.

도 12a에 도시된 것과 같은 트랙정보 관리파일(TRKLIST.MSF)은 시작점에서 32바이트 헤더를 가지고 있다. 헤더는 슬롯들(0x0000)와 (0x0010)을 가지고 있다.

데이터의 다음 부분들은 헤더의 시작점으로부터 순차적으로 배열되어 있다.

\* BLK ID-TL0/TL1 (4 bytes)

이 데이터는 TL0 = 0x544C2D30과 TL1 = 0x544C2D31과 같이 상수이다.

\* T-TRK (2 bytes)

이 데이터는 1에서 400 사이의 값을 가지는 악보의 총 수이다.

\* MCode (2 bytes)

이 데이터는 장비의 형태와 기록 장치의 제조업자를 식별하는 제조업자 코드를 나타낸다. 그 코드들은 메모리시트(sheet)(1)내에 데이터를 기록하기 위해 이용되는 기록장치의 제조업자를 식별하기 위한 제어 코드들이다. 그 코드들은 인가자가 허가를 할 때에 주어진다. 장비 형태 코드는 허가를 내 준 회사에 의해 제어된다.

\* REVISION (4 bytes)

이 데이터는 트랙정보 관리파일(TRKLIST.MSF)이 현재까지 수정된 횟수를 나타낸다. 그 수는 기록 동작이 수행될 때마다 증가된다.

\* YMDhms (4 bytes)

이 데이터는 트랙정보 관리파일(TRKLIST.MSF)이 마지막으로 갱신되었던 날짜와 시간을 나타낸다.

\* N1 (OP) (1 byte)

이 데이터는 메모리시트(1)의 분자축 순차 번호이다. 만약 단지 한 개의 부분이 이용된다면, 그 번호는 모두에 대해 0x0이 된다.

OP는 이 데이터가 선택적이라는 것을 나타낸다.

\* N2 (OP) (1 byte)

이 데이터는 메모리시트(1)의 분모축 순차 번호이다. 만약 단지 한 개의 부분이 이용된다면, 그 번호는 모두에 대해 0x0이 된다.

\* MSID (OP) (2 bytes)

이 데이터는 메모리시트(1)의 ID이다. 만약 다수의 세트들이 이용된다면, 단일한 번호(T. B. D.)가 MSID로 이용된다. T.B.D.는 미래에 결정되는 (정의되는) 구문의 축약이다.

\* S-TRK (2 bytes)

이 데이터는 특수 트랙들(401 - 408) (T. B. D.)을 나타낸다. 정상적으로는, 이 데이터가 0x0000의 값을 가진다.

\* PASS (op) (2 bytes)

이 데이터는 암호(T. B. D.)이다.

\* APP (OP) (2 bytes)

이 데이터는 재생 응용(T. B. D.)을 나타낸다. 정상적으로는 0x0000의 값을 가진다.

\* INF-S (OP) (2bytes)

이 데이터는 오브젝트로 취급되는 전체 메모리시트(1)를 가지는 추가정보를 가리키는 포인터이다. 만약 이러한 추가정보가 없다면, 이 포인터는 00에서 설정된다.

INS-F는 본 발명의 제2 포인터들을 참고하는 포인터를 중 한 개다.

\* S-YMDhms (OP) (4 bytes)

이 데이터는 높은 정확도를 가지며, 날짜와 시간을 기록할 수 있는 장치에 의해 트랙정보 관리파일(TRKLIST.MSF)이 갱신된 날짜와 시간이다. 상술한 헤더 뿐만아니라, 트랙정보 관리파일(TRKLIST.MSF)의 제1 클러스터의 마지막 16 바이트에 기재되어 있으며, 헤더의 것과 동일한 BLK ID-TLO, MCode와 REVISION이 기술되어 있다.

더욱이, 도면에 도시된 바와 같이, 헤더의 것과 동일한 BLK ID-TLO, MCode와 REVISION은 제2 클러스터의 제1 과 마지막 슬롯들내에 기술되어 있다.

그들은 다음과 같은 기능들을 가지고 있다.

오디오 장치에서는, 메모리시트(1)가 제거되거나 또는 전원이 기록 동작중에 중단된다. 이러한 경우에는, 동작이 재개될 때에 장애(abortion)의 발생을 인식하는 것이 필요하다. 상술한 바와 같이, REVISION은 블록의 시작과 끝에 기술된다. 블록의 데이터가 갱신될 때마다, REVISION의 값은 1씩 증가한다. 만약 장애가 블록을 갱신할 때에 발생한다면, 블록의 시작에서 REVISION의 값은 동일한 블록의 마지막에서의 REVISION 값과 다르게 된다. 이러한 차이는 장애가 발생했다는 것을 나타낸다. 트랙정보 관리파일(TRKLIST.MSF)은 백업파일을 가지고 있으므로, 트랙정보 관리파일(TRKLIST.MSF)을 바로 이전의 수정으로 되돌리는 것이 쉽다. 이와 같이, 다수의 데이터 항목들을 제공함으로써, 장애가 높은 확률로 발견될 수 있다. 장애가 인식될 때에, 여러 메시지와 같은 경고가 발생된다는 것을 주목하자.

더욱이, 상수(BLK ID-TLO, BLK ID-TL1)들은 16K8 블록의 시작위치로 삽입된다. 그러므로, 상수들은 손상된 FAT의 복구에 표준 기준으로 이용될 수 있다. 즉, 각 블록의 시작위치에 기술된 상수들을 살펴봄으로써, 파일의 형태가 식별될 수 있다. 더욱이, 상수(BLK ID-TLO, BLK ID-TL1)들은 블록의 시작과 마지막 모두에서 기술되어 있다. 그러므로, 신뢰성이 또한 검사될 수 있다.

ATRAC3 데이터를 기록하기 위한 데이터 파일은 트랙정보 관리파일(TRKLIST.MSF)과 비교해봤을 때에 매우 큰 저장 용량을 가지고 있다. 예를 들면, 데이터 파일은 연속된 수 천 개의 블록들을 가지고 있다. 데이터 파일에 대해서는, BLOCK SERIAL이라고 불리는 블록 번호가 나중에 기술될 것이다. 정상적으로는 다수의 데이터 파일들이 메모리시트(1)내에 존재한다는 것을 주목하자. 그러므로, 나중에 기술될 정보 내용(CONNUMO)을 이용하여 내용들의 식별 뿐만아니라 블록 번호(BLOCK NUMBER)가 할당이 되어 있지 않다면, FAT가 손상된 경우에 복구하는 것이 어렵다. 마찬가지로, 틀린 논리에 의해 발생된 파일 문제는 FAT 자체가 손상되지 않았더라도 발생하게 된다. 그러나, 제조업자 코드(MCode)가 블록의 시작과 마지막에 기술되어 있으므로, 기록된 제조업자의 장비 형태가 식별되게 된다.

트랙정보 관리파일(TRKLIST.MSF)에서는, 헤더 다음에 트랙에 관한 정보를 기술하고 있는 트랙 정보영역(TRKINF)과 트랙의 각각에 있는 한 부분 또는 악보들에 관한 정보를 기술하는 부분 정보영역(PRTINF)이 구성되어 있다.

자세히 설명하자면, 헤더 다음에는, 데이터 파일로서 트랙에 관한 트랙 정보영역(TRKINF - 001)과 부분 정보영역(PRTINF - 001) 및 다른 트랙에 관한 트랙 정보영역(TRKINF - 002)과 부분 정보영역(PRTINF - 002)등이 기술되어 있다.

트랙 정보영역(TRKINF - 001)과 부분 정보영역(PRTINF - 001)의 상세한 부분은 도 12b에 도시되어 있

다. 여기에서 기술된 데이터의 부분들은 다음과 같이 순차적으로 기술될 것이다.

\* T0 (1 byte)

이 데이터는 상수 (0x74의 T0)이다.

\* LT (1 byte)

이 데이터는 재생 한계의 존재 또는 비존재를 나타낸다. 0x80의 LT값은 재생 한계의 존재성을 나타내고, 0x00의 LT값은 재생 한계의 비존재성을 나타낸다. 다른 값들은 재생 동작이 금지되었다는 것을 나타낸다.

\* INF - nnn (0P) (2 bytes)

이 데이터는 트랙의 추가정보를 가리키는 포인터이다. 그 포인터는 나중에 기술될 추가정보파일 (INFLIST.MSF)내에 있는 색인 포인터들 (INF-001 ~ INF-409)을 명시하는 범위 (000 ~ 409)내의 값을 가진다. 값 (000)은 어떤 색인 포인터도 가리키지 않는다는 것을 주목하자. 즉, 000의 포인터를 가지는 트랙은 추가정보와는 관련이 없다.

INF-nnn은 본 발명에 의해 제공된 제2 포인터로 참조되는 포인터이다.

\* FNM-nnn (4 bytes)

이 데이터는 ATRAC3 데이터를 저장하기 위해 데이터 파일에 할당되어 있으며, 범위 (0x0000 ~ 0xFFFF)내의 값을 가지는 파일번호이다. 이 데이터는 데이터 파일을 (A3Dnnnn.MSA)내에 포함된 ASCII 문자인 nnnn을 16진 형태인 0xnnnn으로 변환한 결과로서 얻어진다.

\* CONTENTS KEY-nnn (8 bytes)

이 데이터는 각 내용에 대해 생성된 특수 값이며 메모리시트(1)의 보안 블록내에서 암호화된다.

\* S-SAM (D) SERIAL - nnn (16 bytes)

이 데이터는 메모리시트(1)의 데이터를 복구하는데 이용되는 장치에 대해 고유하게 매겨진 일련 번호이다.

\* APP CTL (0P) (4 bytes)

이 데이터는 응용 변수(T. B. D.)이며, 일반적으로 0x0000의 값을 가진다.

\* CONNUM-nnn (4 bytes)

이 데이터는 각 내용에 대해 생성된 내용 축적(cumulative) 번호이다. 내용 축적 번호는 기록 장치의 보안 블록내에서 보존되므로 한 개의 메모리시트내에서는 어떤 축적 번호도 중복되지 않는다.

\* P-nnn (2 bytes)

이 데이터는 악보(데이터 파일)를 작곡하는 부분들의 번호이다. 그 번호는 범위 (1~2,309)내에 있는 값을 가진다.

\* XT (0P) (2 bytes)

이 데이터는 아래에 기술되는 INX에 의해 표현되는 포인터로부터 시작되는 재생 시간(SU)이다. 0x0000의 값은 아무런 설정도 의미하지 않으며, 0xFFFF의 값은 악보의 마지막을 의미한다.

\* INX-nnn (0P) (4 bytes)

이 데이터는 악보내의 특정 부분을 가리키는 포인터이다. 그 포인터는 악보의 시작과 관련되어 SU로 표현된다. 그러므로, 악보의 시작으로부터 10초 동안에 해당되는 부분을 이용자가 들을 수 있게 하는 종래의 음악 스캔 기능이 개선되며, 특정 부분은 명시될 수 있다.

\* YMDhms-S (4 bytes)

이 데이터는 재생 한계를 가지는 트랙을 재생시키는 동작이 시작될 수 있는 날짜와 시간이다. 0x00000000은 이 데이터가 이용되지 않는다는 것을 의미한다.

\* YMDhms-E (4 bytes)

이 데이터는 재생 한계를 가지는 트랙을 재생시키는 동작이 종료되는 날짜와 시간이다. 0x00000000의 값은 이 데이터가 사용되지 않는다는 것을 의미한다.

\* MT (1 byte)

이 데이터는 재생 한계를 가지는 트랙을 재생시키는 동작이 수행될 수 있는 횟수이다. 0x00의 값은 이 데이터가 사용되지 않는다는 것을 의미한다.

\* CT (1 byte)

이 데이터는 재생 한계를 가지는 트랙을 재생시키는 동작이 수행되었던 횟수이다. 0x00의 값은 이 데이터가 사용되지 않는다는 것을 의미한다.

\* CC (1 byte)

이 바이트 데이터는 복사 제어를 위해 이용된다. 자세히 말하자면, 00, 01과 10의 값은 복사의 금지를 의미한다. 단지 복사는 단지 제1 발생과 복사의 자유(freedom of copying)를 각각 발생시키기 위

해 허용된다. "제1 발생을 위해서만 복사가 허용됨"의 경우에는, 발생한 제1 발생의 복사를 하는 동작이 금지된다는 것에 주목하자.

\* CN (1 byte)

이 바이트 데이터는 복사들의 횟수이다. 00의 값은 복사 금지를 의미하며, 0xFF의 값은 제1 발생이 생성될 때까지의 무한대의 복사들의 횟수가 발생할 수 있다는 것을 의미한다. 01~0xFE 범위내의 값은 재생가능한 복사들의 횟수를 나타낸다. 복사가 수행될 때마다, 카운터는 증가하게 된다.

상술한 64 바이트들의 영역은 트랙 정보영역(TRKINF-nnn)이며, 도 12b에 도시된 16 바이트들의 마지막 영역은 부분 정보영역(PRTINF-nnn)이다.

부분 정보영역(PRTINF-nnn)은 다음과 같은 순서대로 배열된 한 부분에 관한 정보를 기록하기 위해 이용된다.

\* PR (1 byte)

이 정보는 상수(0x50의 PR)이다.

\*A-nnnn (2 byte)

이 정보는 그 부분의 특성들(attributes)이다. 자세히 설명하자면, 특성들은 1바이트 모드와 1바이트의 SCMS(Serial Copy Management System) 정보이다. 자세한 것은 나중에 기술될 것이다.

\* PRSIZE-nnnn (4 bytes)

이 정보는 그 부분의 2 바이트 클러스터 크기, 1바이트 시작 SU와 1바이트 마지막 SU이다.

\* PRKEY-nnnn (8 bytes)

이 정보는 숫자 데이터를 암호화하는 블록키를 생성하기 위해서 내용키와 함께 한 쌍으로 이용되는 키이다. 그것의 초기값은 00이다. 한 부분이 편집 동작에서 발생할 때마다, 이 키는 1씩 증가한다.

상술한 2 바이트 A-nnnn은 도 13과 도 14에 도시된 정보를 기록하기 위해서 하위 바이트와 상위 바이트로 구성된다.

먼저, A-nnnn의 하위 바이트는 도 13에 도시된 ATRAC3 시스템의 모드를 나타내는 모드 정보를 규정하는데 이용된다.

도 13은 64MB 메모리시트에 대한 기록시간, 데이터 전송율과 6개의 모드들, 즉 HQ, SP, CD, LP1, LP2와 모노(mono)의 각각에 대한 압축성(compressibility)과 1 SU내의 바이트들의 수를 도시한 표이다.

상위 바이트내에 기술된 정보는 도 14에 도시되어 있다. 도면에 도시된 바와 같이, 비트 0은 엠파시스(emphasis) 기능이 동작중인지 아닌지를 나타내는 엠파시스 비트(emphasis bit)이다. 비트 1은 정상 재생 또는 재생 스킵(skip)을 나타내는 재생-스킵 비트이다. 비트 2는 발생한 FAX등에 대한 오디오 데이터 또는 음성 데이터를 나타내는 정보를 표현하는 데이터 세그먼트 비트이다. 비트 3과 4는 예비 비트들이다.

비트 5와 6의 결합들은 도면에 도시된 SCMS 정보를 기술하는데 이용된다.

비트 7은 입력 동작이 허용되었는지 또는 금지되었는지를 나타내는 입력보호 비트이다.

위에서 기술된 것은 트랙정보 관리파일(TRKLIST.MSF)내의 데이터 파일을 제어하기 위한 변수들을 기술하기 위한 영역이다.

상술한 바와 같이, 트랙정보 관리파일(TRKLIST.MSF)의 나머지 32-KB 영역은 NAME1과 NAME2를 기술하는데 이용된다.

도 15는 1바이트 코드들로 이루어진 정보를 기술하기 위한 블록인 NAME1의 상세한 데이터 구조를 도시한 도면이다. 나중에 기술될 NAME1과 NAME2는 트랙정보 관리파일(TRKLIST.MSF)내의 시작에서부터 8바이트들의 배수의 오프셋에 있는 8바이트들로 구성된다. 8바이트 단위는 슬롯이라고 부른다. NAME1의 시작은 헤더를 기술하는데 이용되는 슬롯 0x8000이다. 시작 슬롯 0x8000 다음에는 이름과 포인터들이 온다. NAME1의 마지막 슬롯은 헤더와 같이 동일한 데이터를 기술하는데 이용된다.

\* BLK ID-NM1 (4 bytes)

이 데이터는 블록의 내용들을 식별하는 상수(0x4E4D2D31의 NM1)이다.

\* MCode (2 bytes)

이 데이터는 장비의 형태와 제조업자를 식별하는 코드이다.

\* PNM1-nnn, PNM1-S (OP) (4 bytes)

이러한 데이터는 1바이트 코드 NM1을 가리키는 데이터이다. 자세히 설명하자면, PNM1-S는 메모리시트(1)를 나타내는 이름을 가리키는 포인터이다. 한편, PNM1-nnn(nnn = 1에서 408 까지)은 음악의 이름을 가리키는 포인터이다.

각 포인터들은 블록내의 2 바이트의 시작 위치, 2 비트들을 차지하는 문자 코드의 형태와 14 비트를 차지하는 데이터의 크기로 구성되어 있다.

시작 위치는 NM1 영역의 시작과 관련된 오프셋이다. 바이트 형태로 표현한다면, 시작 위치는 범위 0x000 - 0x3989내의 값을 가진다.



문자 코드의 형태는 ASCII 형태를 나타내는 0의 값과, 'ASCII + 일시적인 이름'을 나타내는 1의 값 또는  
정정된 8859-1 형태를 나타내는 2의 값을 가진다.

14 비트를 차지하는 데이터의 크기는 문자 데이터와 0x00의 마지막 바이트의 크기이다. 이 크기는  
범위 0x000- 0x398c내의 값을 가진다.

\* NM1-nnn (0P)

이 데이터는 음악의 이름 또는 메모리시트(1)의 이름을 나타내는 1바이트 코드로 표현된 가변길이 정보  
이다. 이름은 0x00의 마지막 코드에 의해 종료된다.

도 16은 NAME2의 상세한 데이터 구조, 즉 2 바이트 코드로된 정보를 기술하는 영역을 도시한 도면이  
다.

NAME2의 시작은 헤더를 기술하는데 이용되는 슬롯 0xC000이다.

슬롯 0xC000의 시작은 헤더와 같이 동일한 데이터를 기술하는데 이용된다.

\* BLK ID-NM2 (4 bytes)

이 데이터는 블록의 내용들을 식별하는 상수(0x4E4D2D32의 NM1)이다.

\* MCode (2 bytes)

이 데이터는 장비의 형태와 제조업자를 식별하는 코드이다.

\* PNM2-nnn, PNM2-S (0P) (4 bytes)

이러한 데이터는 2 바이트 코드 NM2를 가리키는 데이터이다. 자세히 설명하자면, PNM2-S는 메모리시  
트(1)를 나타내는 이름을 가리키는 포인터이다. 한편, PNM2-nnn(nnn = 1에서 408 바이트)은 음악의  
이름을 가리키는 포인터이다.

각 포인터들은 블록내의 2 바이트의 시작 위치, 2 비트를 차지하는 문자 코드의 형태와 14 비트를 차  
지하는 데이터의 크기로 구성되어 있다.

시작 위치는 NM2 영역의 시작과 관련된 오프셋이다. 바이트 형태로 표현한다면, 시작 위치는 범위  
0x000 ~ 0x3987내의 값을 가진다.

문자 코드의 형태는 일본어(MS-JIS)를 나타내는 0의 값과, '한국어(KS C5601 -1989)를 나타내는 1의 값  
또는 중국어(GB2312 -80)를 나타내는 2의 값을 가진다.

14 비트를 차지하는 데이터의 크기는 문자 데이터와 0x0000의 마지막 바이트의 크기이다. 이 크  
기는 범위 0x000- 0x398c내의 값을 가진다.

\* NM1-nnn (0P)

이 데이터는 음악의 이름 또는 메모리시트(1)의 이름을 나타내는 2 바이트 코드로 표현된 가변길이 정보  
이다. 이름은 0x0000의 마지막 코드에 의해 종료된다.

4-5. 데이터 파일 (A3Dnnnnn.MSA)

다음에는, ATRAC3을 저장하는 데이터 파일 (A3Dnnnnn.MSA)의 구조가 기술된다.

도 17은 N 바이트로 된 SU에 대한 데이터 파일 (A3Dnnnnn.MSA)내에 저장된 데이터의 배열 블록을 도시  
하고 있다.

이 파일은 8바이트를 각각 가지는 슬롯들로 구성되어 있다. 도 17에 도시된 슬롯들은 0x0000에서 시  
작되며, 0x3FF0에서 종료한다.

파일의 시작에 있는 4개의 슬롯들은 아래에 기술될 데이터를 포함하는 헤더이다. 제2 슬롯은 BLOCK  
SEED이다. 마지막 슬롯의 바로 이전에 있는 슬롯은 또한 BLOCK SEED이다. 마지막 슬롯은 BLK ID-  
A3와 MCode이다.

\* BLK ID-A3D (4 bytes)

이 데이터는 블록의 내용을 식별하는 상수(0x41324420의 A3D)이다.

\* MCode (2 bytes)

이 데이터는 장비의 형태와 제조업자를 식별하는 코드이다. 그 파일이 편집된다면, 이 코드를 갱신하  
는 것이 필요하다.

\* BLOCK SEED (8 bytes)

이 데이터는 암호화에 요구되는 블록키를 생성하는데 이용된다. 블록 시드의 시작 값은 기록 장치의  
SAM(Security Application Module : 50)에 의해 무작위 번호로부터 계산된다. 다음 블록의 블록 시  
드는 시작 값을 1만큼 증가시킴으로써 증가된다. 에러를 조정하는 기술로서는, 동일한 블록 시드가  
제2 슬롯과 마지막 슬롯의 이전에 있었던 슬롯내에 기록된다. 더욱이, 데이터 파일이 편집되더라도  
블록 시드를 갱신하는 것이 필요하지 않다.

\* CONNUM (4 bytes)

이 데이터 파일은 최초로 얻어진 내용 번호이다. 이 초기 내용 번호는 이미 기술된 트랙정보 관리파  
일(TRKL1ST.MSF)의 CONNUM의 것과 초기에는 동일하다. 만약 데이터 파일이 편집되더라도 내용 번호

를 갱신하는 것이 필요하지 않다.

\* BLOCK SERIAL (4 bytes)

이 데이터는 블록의 일련 번호이다. 제1 블록의 일련 번호는 00이며, 다음의 블록 번호는 현재 블록 번호를 1만큼 증가시킴으로써 얻어진다. 데이터 파일이 편집되더라도 일련 번호를 갱신할 필요가 없다.

\* INITIALIZATION VECTOR (8 bytes)

이 데이터는 ATAC3 데이터의 암호화와 복호화에 요구되는 초기값이다. 내용의 시작에서는, 초기값이 00이다. 다음 블록에서는, 마지막 SU의 암호화된 마지막 값과 동일하다. 데이터 파일이 편집되더라도 초기 벡터를 갱신할 필요가 없다.

이러한 헤드 다음에는 순차적으로 배열된 음성 단위인 SU-nnnn가 오게 된다. SU는 1,024 샘플들을 압축한 결과로 얻어진 데이터이다. 데이터의 양은 모드에 따라 변한다. 데이터 파일이 편집되더라도 데이터의 양을 갱신할 필요가 없다.

도 18은 블록당 SU의 수, 데이터의 양, 블록당 마진(margin) 데이터의 양과 각 모드에 대한 데이터 파일의 전송 시간과 전송율을 도시한 표이다.

일반적으로, 64MB를 가진 메모리시트(1)가 이용된다. 이러한 메모리시트(1)의 CD 모드에 대한 데이터를 설명한다. 64 MB를 가진 메모리시트는 3,969 블록들을 가지고 있다. CD 모드에서는, 1 SU의 크기가 320 바이트들이다. 그러므로, 1블록은 (1,024 / 44, 100) 초에 각각 대응하는 52 SU들로 구성되어 있다. 그러므로, 데이터 파일에 대한 전송 시간은 다음과 같다.

$$(1,024/44,100) * 51 * (3,968 - 156) = 4,680 \text{ 초} = 78 \text{ 분}$$

전송율은,

$$(44,100/1,024) * 320 * 8 = 110,250 \text{ bps}$$

4-6. 추가정보파일(INFLIST.MSF)

다음에는 추가정보파일(INFLIST.MSF)이 설명된다.

도 19a는 추가정보파일(INFLIST.MSF)의 구체적인 데이터 구조를 도시한 도면이다.

트랙정보 관리파일(TRKIST.MSF)과 같이, 추가정보파일(INFLIST.MSF)은 16 바이트를 각각 가지는 슬롯들로 구성되어 있다.

추가정보파일의 시작에 있는 슬롯 0x0000은 헤더이다. 헤더 뒤에는 헤더의 설명뒤에 기술될 데이터와 포인터들이 오게 된다.

헤더는 다음과 같은 데이터로 구성된다.

\* BLK ID-INF (4 bytes)

이 데이터는 블록의 내용들을 식별하는 상수(0x494E464F의 INF)이다.

\* T-DAT (2 bytes)

이 데이터는 범위 0~49내의 데이터의 총 수이다.

\* MCode (2 bytes)

이 데이터는 제조업자와 장비의 형태를 식별하기 위한 코드이다.

\* YMDhms (4 bytes)

이 데이터는 기록 날짜와 기록 시간이다.

\* INF-nnn(4 bytes)

이 데이터는 나중에 기술될 추가정보의 단위를 가리키는 색인 포인터이다. 색인 포인터에 의해 지시된 추가정보의 단위들은 슬롯들(또는 2 바이트 단위)로 표현된 가변 길이를 가지고 있다.

INF-001에서 INF-409의 표현은 색인 포인터들을 표시한다.

색인 포인터를 INF-001에서 INF-409까지는 각각 본 발명에 의해 제공된 제1 포인터로 언급된다는 것을 주목하자.

4 바이트의 가변 길이를 가지는 색인 포인터 INF-nnn는 도 19b에 도시된 구조를 가지고 있다. 자세히 설명하자면, 16 개의 최상위 비트들은 포인터이며, 15개의 최하위 비트들은 데이터 크기이다. 포인터와 데이터 크기 사이에 있는 1 비트는 무효 플래그(invalidity flag)이다.

포인터에 의해 지시되는 것은 한 단위의 추가정보의 시작 위치이다. 16 개의 최상위 비트에 의해 표현되는 포인터는 범위 0x0000~ 0xFFFF내의 값을 가진다. 나중에 기술될 한 단위의 추가정보는 Dataslot-0000~Dataslot-FFFF를 이용하여 기술된다. 포인터는 바이트 주소 0x8000에 위치한 Dataslot-0000에 대한 오프셋이다. 즉, 포인터는 0x8000에서의 데이터 슬롯 0000에 대한 오프셋에 있는 슬롯 위치를 가리키거나 또한 오프셋은 포인터에 의해 표현된다. 데이터 크기는 음악에 대한 추가정보를 나타내는 데이터의 총량이다.

자세히 설명하자면, 15 개의 최하위 비트들내에 기술된 데이터 크기는 슬롯들의 총 수이다. 한 단위

의 추가정보는 한 슬롯의 시작으로 부터 시작하고 추가정보에 대한 마지막 슬롯내의 한 지점에서 종료한다. 그 지점과 마지막 슬롯의 종료점 사이의 영역은 00으로 채워진다.

색인(index) 포인터의 무효 플래그는 색인 포인터에 의해 지시된 한 단위의 추가정보가 유효한지 또는 무효한지를 나타내는 식별정보이다. 0의 무효 플래그는 추가정보의 유효 단위를 나타내며, 1의 무효 플래그는 추가정보의 무효 단위를 나타낸다.

INF-409에 의해 표시된 제1 색인 포인터는 전체적으로 앨범에 대한 추가정보를 지시한다.

이미 설명한 바와 같이, 실제 추가정보를 포함하는 추가정보의 단위는 한 개의 데이터 슬롯 또는 기술(description)의 단위로서 이용되는 슬롯을 가지는 추가정보파일의 데이터 슬롯들 0000 - FFFF내에 기술된다.

도 20은 추가정보 단위의 데이터 구조를 도시한 도면이다.

도면에 도시된 바와 같이, 추가정보의 단위는 단위의 시작에서 8바이트의 헤더를 포함한다.

\* IN (1 byte)

이 데이터는 상수(0x69의 IN)이다.

\* ID (1 byte)

이 정보는 추가정보의 형태를 식별한다. 많은 형태의 추가정보가 있다. 이 정보는 키 ID로 칭하며, 그것을 아래에 기술할 SID(sub-ID)와 구별되게 한다.

\* SID (1 byte)

T, B, 0, 1의 SID는 형태를 나타낸다.

\* SIZE (2 bytes)

이 정보는 ID에 의해 표시된 추가정보의 단위의 크기를 나타내는 슬롯들의 수이다. 단위의 크기는 범위 1 ~ 7 FFFF내의 값을 가진다. 크기의 최상위 비트(MSB)는 무효 플래그이다. 자세히 설명하자면, 0의 MSB 비트는 추가정보의 유효 단위를 표현하며, 1의 MSB 비트는 추가정보의 무효 단위를 표현한다.

\* MCode (2 bytes)

이 정보는 장비 업자의 기록코드이다.

이러한 헤더 다음에는 가변 길이를 가진 데이터가 온다. 그 데이터는 실제 추가정보의 핵심이다.

도 21은 대표적인 추가정보를 도시한 도면이다.

0x8xxx의 SIZE 값은 데이터가 삭제되었거나 또는 무효하다는 것을 나타낸다. 추가정보의 형태는 키 ID와 헤더에 기술되어 있는 SID에 의해 표시된다. 이러한 코드들의 상세한 점들은 키 ID와 SID가 아직 정의되지 않았으므로 기술되지 않는다.

추가정보는 ISRC(International Standard Recording Code)라고 부르는 저작권 코드, 작사자와 예술가의 이름과 같은 음악 정보와, 하드웨어 제어정보를 포함한다. 음악 정보의 경우에는, 2 바이트가 음악의 문자 코드를 나타내기 위해 데이터의 시작점에 추가된다.

도 22는 추가정보의 단위의 데이터 구조를 도시한 도면이다. 추가정보의 몇 가지 예들이 이러한 데이터 구조를 참조하여 설명된다.

도 23은 시간 스탬프(time stamp)인 추가정보의 데이터 구조를 도시한 도면이다. 도 21에 도시한 바와 같이, 시간 스탬프는 기록시간 스탬프이다.

시간 스탬프의 데이터는 YMDhms 형태로 저장된다. 전체 슬롯이 시간 스탬프에 의해 차지되지 않았다면, 나머지 영역은 00으로 채워진다.

도 24는 재생 로그 파일인 추가정보의 데이터 구조를 도시한 도면이다. 도면에 도시된 바와 같이, YMDhms 데이터는 로그된다(logged). 이 때에 YMD는 각각 연도, 달과 날짜를 표시하며, hms는 각각 시간, 분과 초를 나타낸다.

도 25는 예술가의 이름, ISRC와 TOCID를 포함하는 추가정보의 데이터 구조를 도시한 도면이다. 이 보기에서는, 예술가의 이름이 1바이트 코드로 슬롯내에 기술되어 있다. 슬롯내의 나머지 영역들은 00으로 채워진다. 다음 슬롯은 데이터로 저장된 ISRC 코드이다. 마지막 슬롯은 TOCID의 데이터이다. 도 25에 도시된 추가정보가 삭제된다고 가정한다. 이 경우에는, 도 15에 도시된 추가정보도 도 26에 도시된 정보로 된다. 자세히 말하자면, SIZE가 8xxxx로 갱신된다.

## 5. 파일 기록 과정

다음의 기술은 도 27에 도시된 흐름도를 참조하여 기술된 구성을 가지는 메모리시트(1)내에 오디오 데이터와 추가정보를 기록하는 구동 장치(20)에 수행되는 과정을 설명한다.

도 27에 도시된 바와 같이, 흐름도는 단계(F101)에서 시작한다. 이 단계에서는, 이용자에 의해 수행된 기록동작에 따라, 디지털 입력단자(27), 마이크로폰 입력단자(20), 라인 입력단자(26) 또는 USB 커넥터(28)를 통해 CPU(41)가 구동장치(20)에 공급된 오디오 신호를 메모리시트(1)내에 데이터 파일로 기록하는 과정을 수행한다.

자세히 설명하자면, CPU(41)는 오디오 신호에 대한 필요한 과정을 수행하기 위해서 DSP(49)와 같은 입력

시스템과 구성 요소들을 각각 구동시키며, 처리된 오디오 데이터를 RAM(41b)내에 저장한다.

CPU(41)는 메모리 인터페이스(42)를 통해 SAM(50)을 이용하여 메모리시트(1)를 검증하는 과정을 수행한다. 검증 과정이 성공적으로 종료되었을 때에, RAM(41b)내에 저장된 오디오 데이터는 암호화되며, 암호화의 결과는 메모리시트(1)로 공급되어 한 개의 데이터 파일 또는 다수의 데이터 파일들(트랙들)로 기록되어진다.

상술한 한 개의 데이터 파일 또는 다수의 데이터 파일들을 기록하는 과정이 완료되면, 과정의 흐름은 트랙정보 관리파일(TRKLIST.MSF)이 갱신되는 단계(F102)로 진행한다.

트랙정보 관리파일(TRKLIST.MSF)을 갱신하기 위해서, CPU(41)는 메모리시트(1)내에 기록된 데이터 파일들의 각각에 대한 RAM(41b)상의 트랙 정보영역(TRKINF)과 부분 정보영역(PRTINF)내에 저장되는 데이터를 발생시킨다. 그리하여 도 12에 도시된 트랙정보 관리파일(TRKLIST.MSF)의 데이터를 재구성하게 된다. 재구성된 트랙정보 관리파일(TRKLIST.MSF)은 메모리시트(1)내에 공급되어 메모리시트(1)내에 이미 존재하는 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다.

과정의 흐름은 단계(F103)로 진행한다. 이 단계에서는, 메모리시트(1)내에 기록된 데이터 파일들과 관련된 추가정보를 입력시키는 과정이 수행된다.

추가정보의 보기들은 음악의 이름과 편집과정에서 이용자에 의해 공급된 예술가의 이름 또는 오디오 신호와 함께 구동장치(200)에 공급된 여러 가지 종류들의 정보이다. 오디오 신호가 디지털 입력단자(27) 또는 USB 커넥터(28)를 통해 구동장치(20)에 공급된 디지털 데이터라고 가정한다. 이 경우에서는, 여러 종류의 정보가 오디오 신호와 함께 전송된다. 추가정보는 전송된 정보에 근거하여 발생될 수 있다.

추가정보가 이용자 또는 상술한 외부장치로부터 수신될 때에, CPU(41)는 새로운 추가정보파일(INFLIST.MSF)을 발생시키거나 또는 추가정보를 포함하고 있는 기존의 추가정보파일(INFLIST.MSF)을 RAM(41b)내의 추가정보단위로 갱신한다. 그 때에는, 추가정보의 단위에 대해 도 19에 도시된 색인 포인터(INF-*nnn*)가 설정된다는 것은 말할 필요도 없다.

추가정보파일(INFLIST.MSF)이 발생된 후에, 과정의 흐름은 단계(F104)로 진행한다. 이 단계에서는, CPU(41)가 추가정보파일(INFLIST.MSF)에 대한 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다. 이 경우에서는, 색인 포인터에 의해 지시된 추가정보파일(INFLIST.MSF)내의 추가정보의 한 단위 또는 다수의 기록된 데이터 파일들을 결합시키기 위해, 도 12b에 도시된 포인터들(INF-*nnn*)이 기록된 데이터 파일들에 대응하는 트랙정보영역(TRKINF)내에 설정된다.

이와 같이 재구성된 트랙정보 관리파일(TRKLIST.MSF)은 메모리시트(1)로 공급되어 메모리시트(1)내에 이미 존재하는 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다.

과정의 흐름은 단계(F105)로 진행한다. 이 단계에서는, 단계(F103)에서 발생된 추가정보파일(INFLIST.MSF)이 메모리시트(1)에 공급되어 메모리시트(1)에 추가정보파일(INFLIST.MSF)을 기록하거나 또는 메모리시트(1)내에 이미 존재하는 추가정보파일(INFLIST.MSF)을 갱신한다.

대표적인 음악의 데이터 파일들을 기록하고, 추가정보파일(INFLIST.MSF)을 기록 또는 갱신하고, 그리고, 트랙정보 관리파일(TRKLIST.MSF)을 기록 또는 갱신함으로써, 한 개의 음악 또는 다수의 음악들이 데이터 파일들을 한 개의 파일로서 각각 기록될 수 있으며 적절하게 제어된다. 더욱이, 부수적인 추가 정보도 또한 기록될 수 있다.

상술한 설명으로부터 추가정보 또는 엄밀히 말하자면, 데이터 파일과 각각 결합된 추가정보의 단위들은 다음과 같이 제어된다는 것을 알 수 있다. 추가정보의 한 단위는 추가정보파일(INFLIST.MSF)내의 색인 포인터(INF-*nn*)에 의해 지시된다. 더욱이, 색인 포인터(INF-*nn*)는 트랙정보 관리파일(TRKLIST.MSF)내에 위치해 있으며, 데이터 파일과 결합되어 있는 트랙정보영역(TRKINF)내의 포인터(INF-*nnn*)에 의해 지시된다.

상술한 바와 같이, 추가정보는 2단 구성으로 조직되어 있는 포인터들에 의해 제어되므로, 추가정보를 편집하는 나중의 과정은 매우 효과적인 과정이 된다.

예를 들면, 큰 크기를 갖는 추가정보파일(INFLIST.MSF)을 갱신할 필요가 없이 작은 크기를 가진 추가정보파일(INFLIST.MSF)만을 갱신함으로써 여러 종류의 편집 과정이 수행될 수 있다. 더욱이, 특수 제어도 추가정보파일(INFLIST.MSF)을 갱신시킴으로써 실행될 수 있다. 특수 제어의 보기는 데이터 파일에서 다른 데이터 파일로 추가정보를 전송하고 다수의 데이터 파일들을 추가정보의 한 부분과 결합시키는 것이다.

전체 메모리시트를 위한 추가정보는 데이터 파일을 기록하는 과정과는 상관없이 기록된다.

이용자가 전체 메모리시트를 위한 추가정보를 입력시킬 때에, 도 27에 도시된 흐름도의 단계(F103 ~ F105)의 부분들이 수행된다. 이 경우에서는, 그 때에 갱신된 포인터가 도 12a에 도시된 트랙정보 관리파일(TRKLIST.MSF)내의 포인터(INF-S)로 이용되며, 대표적으로 도 19a에 도시된 색인 포인터들중 색인 포인터(INF-409)는 추가정보파일(INFLIST.MSF)내의 색인 포인터로 이용된다.

상술한 바와 같이, 데이터 파일에 대한 추가정보는 데이터 파일과 같이 기록된다. 기존의 데이터 파일에 대한 추가정보도 물론 기록될 수 있다. 이 경우에서는, 추가정보와 결합되는 기존의 데이터 파일에 대해서는, 도 27에 도시된 흐름도의 단계(F103 ~ F105)의 부분들이 수행된다.

## 6. 파일삭제 과정

다음의 기술은 도 28에 도시된 흐름도를 참조하여 데이터 파일과 결합된 추가정보와 데이터 파일을 삭제하는 과정을 설명한다.

도 28에 도시된 바와 같이, 과정은 단계(F201)에서 시작된다. 이 단계에서는, 이용자가 기록된 데이터 파일을 명시하고 명시된 데이터 파일을 삭제하는 동작을 수행할 때에 CPU(41)가 삭제 명령을 수신한다. 다음 단계(F202)에서는, 삭제될 데이터 파일에 대한 트랙정보영역내의 포인터(INF-nnn)가 RAM(41b)내에 이미 로드되어 있던 메모리시트(1)의 트랙정보 관리파일(TRKLIST.MSF)로부터 판독된다.

다음 단계(F203)에서는, 추가정보파일(INFLIST.MSF)이 메모리시트(1)로부터 판독된다. 다음 단계(F204)에서는, 단계(F202)에서 판독된 색인 포인터(INF-nnn)에 의해 지시되는 추가정보파일(INFLIST.MSF)내의 색인 포인터(INF-nm)가 무효 플래그가 동작되어 있는 내용으로 즉, 도 19b에 도시된 바와 같이 무효 플래그가 '1'로 설정되어 있는 내용으로 갱신된다.

다음 단계(F205)에서는, CPU(41)가 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다. 자세히 설명하자면, 지워질 데이터 파일에 대한 트랙정보영역(TRKINF-xxx)과 부분정보영역(PRTINF-xxx)이 삭제된다. 이와 같이, 데이터 파일의 삭제가 실행된다.

트랙정보영역(TRKINF-xxx)과 부분정보영역(PRTINF-xxx)을 삭제함으로써, 트랙정보영역(TRKINF-xxx)과 부분정보영역(PRTINF-xxx)에 할당된 영역들은 트랙정보 관리파일(TRKLIST.MSF)내의 자유영역이 된다. 자유영역 다음에 오는 다른 데이터 파일들에 대한 트랙정보영역(TRKINF-xxx)과 부분정보영역(PRTINF-xxx)내에 저장된 데이터들은 전방으로 이동할 필요가 있다.

트랙정보 관리파일(TRKLIST.MSF)이 상술한 바와 같이, 재구성된 후에, 다음 단계(F206)에서는, 재구성된 트랙정보 관리파일(TRKLIST.MSF)이 메모리시트(1)에 공급되어 메모리시트(1)내에 이미 존재하는 트랙정보 관리파일(TRKLIST.MSF)을 갱신한다.

계속해서, 다음의 단계(F207)에서는, 단계(F204)에서 갱신된 추가정보파일(INFLIST.MSF), 즉 단계(F204)에서 무효 플래그가 동작되어 있는 추가정보파일(INFLIST.MSF)이 메모리시트(1)에 공급되어 메모리시트(1)내에 이미 존재하는 추가정보파일(INFLIST.MSF)을 갱신한다.

데이터 파일을 삭제하는 과정은 단계(F207)에서 수행되는 과정에 의해 종료된다.

상술한 과정에서는, 데이터 파일 그 자체의 삭제는, 트랙정보 관리파일(TRKLIST.MSF)을 단지 갱신함으로써 실행된다. 더욱이, 트랙정보 관리파일(TRKLIST.MSF)은 비교적 작은 크기를 가진 파일이기 때문에, 데이터 파일의 삭제와 삭제된 데이터의 재구성은 그렇게 힘든 과정은 아니다.

삭제된 데이터 파일과 결합된 추가정보는 추가정보파일(INFLIST.MSF)내의 결합된 색인 포인터내의 무효 플래그를 단지 동작시킴으로써 삭제된다.

추가정보파일(INFLIST.MSF)의 크기는 그 안에 저장된 추가정보의 양에 따라 증가한다. 삭제 과정에서는, 전체 추가정보파일(INFLIST.MSF)을 재구성하는 것이 필요하지 않다. 대신에, 결합된 색인 포인터만 갱신될 필요가 있다. 즉, 추가정보 그 자체의 단위만이 조절된다. 그러므로, 과정 로드는 증가하지 않는다.

상술한 설명으로부터 데이터 파일과 데이터 파일과 결합된 추가정보는 실시예에 의해 실행된 것과 같은 휴대용 및 작은 구동장치(20), 즉 높은 소비전력이 제공되지 않는 장치에 의해 수행된다는 것을 알 수 있다.

다른 말로 표현하면, 메모리시트(1)내의 추가정보의 관리 시스템은 장치가 문제 없이 적은 소비전력으로 편집작업을 수행하도록 지원한다.

전체 메모리시트(1)에 대한 추가정보가 삭제되는 경우를 생각해 보는 것도 중요하다. 이 경우에는, 도 28에 도시된 흐름도에 의해 표현되는 데이터파일 삭제과정이 전체 메모리시트(1)에 대해 특별히 수행된다. 자세히 말하자면, 단계(F202)에서, 색인 포인터가 얻어지는 트랙정보 관리파일(TRKLIST.MSF)의 헤더내에 있는 포인터(INF-S)로부터 시작된다. 데이터 파일을 삭제시키는 한 과정으로 트랙정보 관리파일(TRKLIST.MSF)을 갱신시키는 단계(F205, F206)에서 수행되는 과정이 이 경우에는 요구되지 않는다는 것을 알 수 있다.

## 7. 추가정보파일의 압축과정

추가정보가 상술한 바와 같이 무효 플래그를 설정함으로써 삭제된 후에, 추가정보파일은 퍼스널 컴퓨터(11)와, 메모리시트(1)와 호환성이 있는(compatible) 스테이셔너리 형태(stationary)의 구동장치와 같은 데이터 처리 전력이 소모되는 장치를 이용함으로써 재구성 또는 압축된다.

상술한 바와 같이 그렇게 높지 않은 전력이 소모되는 구동장치(20)에서는, 추가정보 그 자체의 단위, 즉, 추가정보 그 자체의 내용은 실제적으로 삭제되지 않는다. 결과적으로, 낭비된 영역은 추가정보 파일(INFLIST.MSF)내에 남아있게 된다.

물론, 이러한 낭비된 영역은 추가정보파일(INFLIST.MSF)내에 남아있을 수도 있다. 그러나, 메모리시트(1)에 대한 추가정보파일(INFLIST.MSF)을 퍼스널 컴퓨터(11)를 이용하여 추가정보파일(INFLIST.MSF)로부터 낭비된 영역을 제거하여 압축시키는 것이 바람직하다.

다음의 기술은 도 29에 도시된 흐름도를 참조하여, 일반적으로 퍼스널 컴퓨터(11) 또는 스테이셔너리 구동장치의 소비전력을 이용함으로써 추가정보파일(INFLIST.MSF)을 압축시키는 과정을 설명한다.

도 29는 퍼스널 컴퓨터(11) 또는 스테이셔너리 구동장치와 같은 고소비전력을 가지는 데이터 처리장치의 CPU에 의해 수행되는 과정을 나타내는 흐름도이다.

예를 들면, 스테이셔너리 구동장치는 높은 소비전력을 가지는 데이터 처리장치로서 설명된다. 높은 소비전력을 가지는 이러한 1개 데이터 처리장치의 구성은 도 30에 도시된 구동장치와 동일하다. 차이점은 높은 소비전력을 가지는 데이터 처리장치가 고성능을 가지는 CPU(41)로 구성된 장치로 여겨질 수 있다는 것이다. 이러한 이유 때문에, 도 29에 도시된 흐름도에 의해 표현된 과정을 수행하기 위한

이러한 데이터 처리장치 또는 스테이셔너리 구동장치의 구성은 설명되지 않는다. 과정을 기술하는데 있어서, 고성능을 가진 중앙처리장치는 도 30에 도시된 CPU(41)와 구별되기 위하여 CPU(41H)로 표시된다. 이와 같이, 다른 블록들은 점미사(H)가 있는 표시법에 의해 표시된다.

도 29에 도시된 바와 같이, 이용자가 압축과정을 수행할 수 있는 데이터 처리장치내에서 사용된 착탈(mounting/demounting)기구(22H)상에 있는 메모리시트(1)압축과정 요청을 발생시킬 때에, 추가정보 파일(INFLIST.MSF)을 압축하는 과정을 나타내는 흐름도는 단계(F301)에서 시작하게 된다. 그 과정은 또한 메모리시트(1)가 마운트된 시간에서 자동적으로 수행될 수 있다는 것을 주목하자.

먼저, 단계(F301)에서는, CPU(41H)가 메모리 인터페이스(42H)를 통해 메모리시트(1)로부터 추가정보파일(INFLIST.MSF)을 판독하며, 그 파일을 RAM(41bh)내에 로드한다.

그 후에는, 다음 단계(302)에서, 변수(n)이 단계(F303)의 고장과 다음 단계로 진행하기 전에 1로 설정된다.

다음 단계(F303)에서는 색인 포인터(INF-(n))가 추가정보파일(INFLIST.MSF)로부터 판독된다. 다음 단계(F304)에서는, 색인 포인터(INF(n))가 검사되어 그 안에 포함된 무효 플래그가 동작되는지 안되는지를 판단한다.

무효 플래그가 동작중이라면, 과정의 흐름은 단계(F307)로 진행한다. 이 단계에서는 변수(n)가 검사되어 마지막 색인 포인터(INF-(n))에 의해 지시된 추가정보의 단위가 처리되었는지를 판단하게 된다. 만약 마지막 색인 포인터(INF-(n))에 의해 지시된 추가정보의 단위가 처리되었다면, 과정의 흐름은 단계(F308)로 진행한다. 이 단계에서는, 변수(n)가 1만큼 증가한다. 그 후에는, 과정의 흐름이 단계(F303)로 되돌아간다.

단계(F304)에서 형성된 판단의 결과가 무효 플래그가 비동작 상태라는 것을 나타내면, 한편, 과정의 흐름은 단계(F305)로 진행한다. 이 단계에서는, 다음의 색인 포인터(INF-(n))가 상방향으로 이동된다. 그 후에는, 다음 단계(F305)에서, 이동된 색인 포인터(INF-(n))에 의해 지시된 추가정보의 단위들이 역시 상방향으로 이동된다. 그 후에는, 과정의 흐름이 단계(F307)로 진행한다. 만약 단계(F307)에서 형성된 판단의 결과가 마지막 색인 포인터(INF-(n))에 의해 지시된 추가정보의 단위가 처리되지 않았다는 것을 나타내면, 과정의 흐름은 변수(n)가 1만큼 증가하는 단계(F308)로 진행한다. 그 후에는, 과정의 흐름이 단계(F303)로 진행한다. 과정의 부분들은 단계(F303)에서 단계(F306)까지 반복적으로 수행된다. 그리고, 추가정보의 가장 최근의 마지막 단위가 단계(F306)에서 상방향으로 이동될 때까지 변수(n)는 각 반복과정에서 1만큼씩 증가된다. 단계(F306)에서 상방향으로 이동된 추가정보의 가장 최근의 유효한 마지막 단위에서는, 단계(F306)에서 형성된 판단의 결과가 조만간에 마지막 색인 포인터(INF-(N))에 의해 지시되는 추가정보의 단위가 처리되었다는 것을 나타내게 될 것이다.

도 30a와 도 30b에 도시된 바와 같이, 압축과정은 새로운 압축된 추가정보파일(INFLIST.MSF)을 생성시킨다.

도 30a는 단계(F301)에서 얻어진 대표적인 추가정보파일(INFLIST.MSF)을 도시한 도면이다. 도면에 도시된 바와 같이, 색인 포인터들(INF-001 ~ INF-005)은 추가정보 단위들(INFdata-001 ~ INFdata-005)을 각각 지시한다.

도 28에 도시된 흐름도에 의해 표현된 삭제 과정의 결과로서, 색인 포인터들(INF-002, INF-004)의 무효 플래그들은 추가정보 단위들(INFdata-002, INFdata-004)이 삭제되었다는 것을 나타내기 위해 설정된다.

도 29에 도시된 흐름도에 의해 표현된 압축과정에서는, 색인 포인터들(INF-001 ~ INF-005)의 무효 플래그들이 단계(F303 ~ F308)들로 구성된 루프내에서 변수(n)를 1만큼씩 증가시킴으로써 한 개씩 순차적으로 검사된다. 설정 무효 플래그가 발견된다면, 다음의 색인 포인터들(INF-(n))과 다음의 색인 포인터들(INF-(n))에 의해 지시된 추가정보의 단위들이 상방향으로 이동되며 도 30b에 도시된 새로운 추가정보파일(INFLIST.MSF)을 생성시킨다. 이 도면에 도시된 바와 같이, 유효 색인 포인터들(INF-001, INF-002, INF-003)과 유효 색인 포인터들(INF-001, INF-003, INF-005)에 의해 지시된 추가정보단위(INFdata-001, INFdata-003, INFdata-005)가 새로운 추가정보파일(INFLIST.MSF)내에 있게 된다. 유효 색인 포인터들(INF-001, INF-003, INF-005)은 도 30b에 도시된 바와 같이 새로운 추가정보파일(INFLIST.MSF)내에 색인 포인터(INF-001), 새로운 색인 포인터(INF-003)와 색인 포인터와(INF-005)로 남게 된다. 반면에 추가정보 단위(INFdata-001, INFdata-003, INFdata-005)는 새로운 추가정보파일(INFLIST.MSF)내에 추가정보 단위(INFdata-001), 새로운 추가정보 단위 INFdata-003), 새로운 추가정보 단위 (INFdata-005)로 각각 남게 된다.

단계(F307)에서 형성된 판단의 결과가 마지막 색인 포인터(INF-(n))에 의해 지시된 추가정보의 단위가 처리되었다는 것을 나타내므로, 추가정보파일(INFLIST.MSF)을 압축시키며, 도 29에 도시된 흐름도에 표현된 과정이 완료된다. 이 경우에는, 단계(F303 ~ F308)로 구성된 루프로부터 벗어나 과정의 흐름이 단계(F309)로 진행한다. 도 30a와 30b에 도시된 보기의 경우에는, 추가정보 단위(INFdata-005)를 지시하는 색인 포인터(INF-005)가 처리된 후에, 과정의 흐름이 단계(F309)로 진행한다.

단계(F309)에서는, CPU(41H)가 상기와 같이 생성된 새로운 추가정보파일(INFLIST.MSF)을 메모리시트(1)에 공급하여 메모리시트(1)내에 이미 존재하는 추가정보파일(INFLIST.MSF)을 경신하게 된다.

추가정보파일(INFLIST.MSF)이 경신될 때에, 추가정보파일(INFLIST.MSF)을 압축하는 과정이 완료된다. 도 30a와 도 30b에서 알 수 있듯이, 추가정보파일(INFLIST.MSF)은 동작중인 무효 플래그를 가지는 어떤 색인 포인터(INF-(n))와 그 색인 포인터(INF-(N))가 지시하는 추가정보 단위(INFdata-(n))를 포함하지 않으며, 더 작은 양의 응축된 데이터를 포함한다.

상술한 압축과정을 수행함으로써, 추가정보파일(INFLIST.MSF)이 최적의 데이터 구조로 재구성될 수 있다. 상술한 압축과정은 단지 무효 플래그를 설정시킴으로써 추가정보를 삭제하는 기능을 지원한다. 즉, 도 28내에 도시된 흐름도에 의해 표현된 과정의 결과로 얻어진 추가정보파일(INFLIST.MSF)내의 남비

된 영역은 제거될 수 있다.

퍼스널 컴퓨터(11)에 의해 수행된 압축과정에서는, 구동장치(20)가 압축된 추가정보파일(INFLIST.MSF)을 도 29에 도시된 호를도에 의해 표현된 추가정보파일(INFLIST.MSF)에 대한 과정을 수행하는 퍼스널 컴퓨터(11)로 전송한다. 그 후에는, 퍼스널 컴퓨터(11)에 의해 새롭게 생성된 추가정보파일(INFLIST.MSF)은 구동장치(20)로 다시 전송되어 메모리시트(1)에 저장된다.

본 발명은 한 실시예에 의해 설명되어졌다. 그러나, 본 발명의 범위는 그 실시예에 의해 제한되어 있지 않다. 특히, 상술한 각 동작의 자세한 처리 절차는 수정된 형태로 변화될 수 있다는 것을 주목하자.

더욱이, 본 발명에 의해 제공된 시스템내의 제1 기록 매체로 언급되는 기록매체는 도1에 도시된 시트 메모리(sheet memory)로 한정되어 있는 것은 아니다. 즉, 제1 기록매체는 메모리 칩, 메모리 카드 또는 메모리 모듈과 같은 다른 외부 형태를 가진 고체 메모리가 될 수 있다. 메모리 장치는 플래시 메모리가 되어야 할 필요는 없다. 다른 형태의 메모리 장치가 이용될 수 있다. 더욱이, 본 발명은 고체 메모리 대신에 미니 디스크, DVO(Digital Versatile Disc), 하드 디스크와 CD-R과 같은 디스크 형태의 기록매체를 이용하는 시스템에도 적용된다.

상술한 실시예에서는, 음악과 같은 정보를 기록하기 위한 오디오 데이터 파일의 추가정보가 설명되었다. 오디오 데이터 파일은 단지 보기일 뿐이다. 즉, 추가정보는 파일내에 저장된 트랙 또는 오디오 데이터에만 제한되어 있지 않다. 예를 들면, 본 발명은 이동화상파일, 정지화상 파일과 오디오 데이터 파일과 같은 다른 파일들에도 동일한 방식으로 적용될 수 있다.

### 발명의 효과

상기의 설명에서 알 수 있듯이, 본 발명은 다음과 같은 효과들을 가지고 있다.

먼저, 본 발명에 의해 제공된 기록 매체내에서는, 추가정보파일이 각각 실제 추가정보와 추가정보 단위들중 한 개를 각각 지시하는 제1 포인터들을 기술하고 있는 한 개의 추가정보 단위 또는 다수의 추가정보 단위를 기록하는데 이용된다. 반면에, 트랙정보 관리파일은 전체 기록매체 또는 데이터 파일과 결합되어 있으며 제1 포인터를 중 한 개를 각각 지시하는 제2 포인터들을 기록하는데 이용된다. 즉, 추가정보파일내에 저장된 다수의 제1 포인터들은 제 포인터를 중 한 개와 부가정보의 단위들의 각각을 결합시킴으로써 실제 추가정보의 내용을 각각 기술하고 있는 부가 정보 단위를 제어하는데 이용된다. 한편, 트랙정보 관리파일내에 저장되어 있는 제2 포인터들은 제1 포인터들을 제어하는데 이용된다.

이와 같이, 데이터 파일과 전체 기록매체에 대한 부가정보는 2단 구성으로 조직된 포인터들에 의해 제어되므로, 여러 종류의 편집과장은 비교적 큰 크기를 가지는 부가정보를 갱신할 필요없이 작은 크기를 가진 트랙정보 관리파일만을 갱신시킴으로써 수행된다. 결과적으로 부가 정보를 편집하는 과정은 매우 효과적이 될 수 있다는 효과가 나타나게 된다.

더욱이, 부가정보 파일내에 저장된 제1 포인터들의 각각은 제1 포인터에 의해 지시된 부가 정보의 단위가 유효한지 또는 무효한지를 나타내는 유효 정보를 포함한다. 이러한 유효 정보를 가지면, 부가정보의 실제 단위는 부가정보의 단위를 지시하는 제1 포인터내에 포함된 유효정보만을 갱신시킴으로써 삭제될 수 있다. 즉, 매우 간단한 과정에 의해 부가정보의 삭제를 실행할 수 있는 효과가 나타난다.

본 발명에 의해 제공된 데이터 처리장치는 상술한 제1과 제2 포인터들의 구성에 의해 부가 정보가 제어되는 기록매체내에 입력 부가정보를 부가정보 파일내의 부가정보의 단위로서 기록한다. 더욱이, 부가정보의 단위가 기록될 때에, 제1 과 제2 포인터들도 또한 기록되기 때문에, 부가정보의 기록된 단위는 데이터 파일 또는 전체 기록매체상에 있는 부가정보로서 제어된다. 결과적으로, 상술한 관리 시스템을 가지는 기록매체가 실행되어, 부가정보를 편집하는 과정이 매우 효과적이 되도록 한다.

본 발명에 의해 제공된 데이터 처리장치는 기록매체로부터 부가정보의 단위를 삭제한다. 상기의 삭제 과정은 다음과 같다. 즉, 부가정보내에 저장되어 있으며, 부가정보의 삭제된 단위를 지시하는 제1 포인터내에 포함된 유효정보를 갱신시켜, 유효정보를 포함하는 제1 포인터에 의해 지시된 부가정보의 단위가 삭제되었다는 것을 나타내는 무효 상태로 유효정보를 변경시킴으로써 이루어진다. 그러므로, 부가정보의 삭제는 이와 같이 어려운 과정이 아니다.

상기 특징은 장치가 예를 들면, 단지 휴대용이고 작은 데이터 처리장치라고 하더라도 부가정보를 편집하는 충분한 기능이 장치에 제공된다는 것을 의미한다.

본 발명에 의해 제공된 데이터 처리장치는 현재의 부가정보 파일의 각각의 제1 포인터내에 포함된 유효정보에 따라 부가정보의 무효화된 단위들을 포함하지 않는 새로운 부가정보 파일을 발생시킴으로써 기록매체 위에 기록된 부가정보를 갱신시킬 수가 있다. 예를 들면, 퍼스널 컴퓨터와 같은 비교적 높은 소비전력을 가진 데이터 처리장치는 실제로 큰 크기를 가진 전체의 부가정보 파일을 처리할 수 있는 반면에 이러한 데이터 처리장치는 유효정보에 따라 단지 부가정보를 재구성하는데 이용된다.

결과적으로 부가정보 파일을 낭비된 영역이 없는 데이터 구조를 가지는 파일로 갱신시키는 효과가 나타나게 된다.

### (57) 청구의 범위

청구항 1. 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 한 개 또는 다수의 보조데이터를 기록하는 기록매체에 있어서,

상기 한 개 또는 다수의 메인데이터를 기록하는 메인데이터 기록영역과,

상기 한 개 또는 다수의 보조데이터로 구성된 보조데이터 파일 중 최소한 한 개의 파일을 기록하여 상기 보조데이터의 기록 위치들을 제어하는 보조데이터 기록영역과,

상기 메인데이터 기록영역에 기록된 상기 메인데이터를 제어하고 상기 메인데이터를 상기 메인데이터와 관련된 상기 보조데이터를 포함하는 상기 보조데이터 파일에 링크(link)시키는 링크정보를 기록하는 관리데이터 기록영역으로 구성된 기록매체.

**청구항 2.** 제 1항에 있어서,

상기 보조데이터 기록영역은 상기 한 개 또는 다수의 보조데이터가 유효한지 또는 무효한지를 나타내는 한 개 또는 다수의 보조데이터의 각각에 대한 정보를 포함하고 있는 기록매체.

**청구항 3.** 한 개 또는 다수의 메인데이터, 상기 메인데이터와 관련된 한 개 또는 다수의 보조데이터와 상기 메인데이터를 제어하는 관리데이터를 기록하는 기록매체에 데이터를 기록하는 방법에 있어서,

상기 메인데이터를 상기 기록매체에 기록하는 단계와,

상기 메인데이터의 기록에 근거하여 상기 관리데이터를 기록하는 단계와,

상기 한 개 또는 다수의 보조데이터로 구성된 보조데이터 파일을 상기 기록매체내에 기록하는 단계와,

상기 메인데이터를 상기 보조데이터에 링크시키는 링크정보를 기록하기 위해 상기 관리데이터를 갱신하는 단계로 구성되어 있는 기록방법.

**청구항 4.** 제 3항에 있어서,

상기 보조데이터 파일은 상기 보조데이터의 기록 위치들을 제어하기 위해 한 개 또는 보조데이터와 보조 관리데이터로 구성되는 기록방법.

**청구항 5.** 제 3항에 있어서,

상기 방법은 상기 메인데이터를 상기 기록매체내에 기록하는 단계 이전에 실행된 상기 메인데이터를 인코딩하는 단계를 포함하는 기록방법.

**청구항 6.** 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터 중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체상에 기록된 보조데이터의 무효데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집방법에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 데이터 판독단계와,

상기 관리데이터의 링크정보에 따라 상기 기록매체로부터 상기 보조데이터 파일을 판독하는 파일 판독단계와,

상기 보조데이터가 상기 보조데이터 파일내의 상기 보조 관리데이터에 근거하여 무효한 것이라고 판단되는 경우에, 상기 보조데이터 파일내의 상기 관리데이터에 의해 무효화된 보조데이터와 결합된 관리데이터를 삭제하고 상기 보조데이터 파일로부터 무효라고 판단된 상기 보조데이터를 삭제하는 삭제단계와,

상기 삭제 단계에서 삭제된 불필요한 정보를 가진 상기 보조데이터파일을 상기 기록매체내에 다시 입력시키는 파일 입력단계로 구성되어 있는 편집방법.

**청구항 7.** 제 6항에 있어서,

상기 보조 관리데이터는 최소한 상기 보조데이터 파일내에 저장된 보조데이터가 유효한지 또는 무효한지를 나타내는 유효정보를 포함하며, 상기 삭제 단계에서, 상기 유효정보는 삭제된 보조데이터가 존재하는지 존재하지 않는지를 판단하기 위해 검사되는 편집방법.

**청구항 8.** 제 6항에 있어서,

상기 삭제 단계에서, 유효한 보조데이터를 가진 관리데이터와 상기 유효한 보조데이터만을 새로운 보조 데이터 파일에 복사시킴으로써 불필요한 정보가 삭제되는 편집단계.

**청구항 9.** 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체상에 기록된 보조데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집방법에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 데이터 판독단계와,

상기 관리데이터의 링크정보에 따라 상기 기록매체로부터 상기 보조데이터 파일을 판독하는 파일 판독단계와,

상기 파일 판독단계에서 판독된 상기 보조데이터 파일내에서 삭제된다고 명시되어 있는 보조데이터와 결합된 특정한 한 개 또는 다수의 보조 관리데이터를 재입력시키는 재입력 단계와,

상기 재입력 단계에서 재입력된 상기 불필요한 보조데이터에 대한 명시된 한 개 또는 다수의 보조 관리 데이터를 가진 상기 보조데이터 파일을 상기 기록매체내에 다시 입력시키는 파일 입력단계로 구성되어 있는 편집방법.

**청구항 10.** 제 9항에 있어서,

상기 보조 관리데이터는 최소한 상기 보조데이터 파일내에 저장된 보조데이터가 유효한지 또는 무효한지를 나타내는 유효정보를 포함하며, 상기 재입력 단계에서, 상기 명시된 한 개 또는 다수의 보조 관리



이터는 상기유효정보를 무효상태로 설정함으로써 재입력되는 편집방법.

**청구항 11.** 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록매체에 기록하는 기록매체에 기록장치에 있어서,

상기 메인데이터, 상기 관리데이터 또는 상기 보조 데이터 파일을 상기 기록매체에 기록하는 기록 수단과,

상기 메인데이터를 상기 기록매체에 기록한 사실에 근거하여 상기 기록매체에 저장된 상기 관리데이터를 갱신하는 제 1 갱신수단과,

서브 데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 보조 관리데이터를 생성하고, 보조데이터 중 최소한 한 개를 제어하는 보조 관리데이터를 발생시키는 보조데이터 파일 발생수단과,

상기 메인데이터를 가지는 상기 보조데이터 파일과 결합된 링크정보를 상기 관리데이터중 1개 데이터로서 기록하기 위해 상기 관리데이터를 갱신시키는 제2 갱신 수단으로 구성된 기록장치.

**청구항 12.** 제 11항에 있어서,

상기 장치는 상기 메인데이터를 상기 기록매체에 기록하기 전에 상기 메인데이터를 인코딩하는 인코딩 수단을 추가로 포함하는 기록장치.

**청구항 13.** 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체에 기록된 보조데이터의 무효데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집장치에 있어서,

상기 기록매체로부터 상기 관리데이터를 판독하는 판독수단과,

상기 데이터를 상기 기록매체에 입력시키는 입력수단과,

상기 기록수단에 의해 상기 기록매체로부터 판독된 상기 보조데이터 파일내의 상기 보조 관리데이터에 근거하여 보조데이터 중 무효한 것이 있는지 또는 없는지를 판단하는 판단수단과,

상기 판단수단에 의해 형성된 상기 판단의 결과가 보조데이터 중 무효한 것이 있다고 지시하는 경우에 상기 보조데이터 파일을 편집하는 작업을,

상기 보조데이터 파일내에 있는 상기 보조 관리데이터에 의해 무효화된 보조데이터와 결합된 관리데이터를 삭제하는 단계와,

상기 보조데이터 파일로부터 무효라고 판정된 상기 보조데이터를 삭제하는 단계와,

상기 편집된 보조데이터 파일을 상기 기록매체내에 입력시키는 상기 입력수단을 제어하는 단계를 실행시킴으로써 제어할 수 있는 제어수단으로 구성된 편집장치.

**청구항 14.** 제 13항에 있어서,

상기 보조 관리데이터는 최소한 상기 보조데이터 파일내에 저장된 보조데이터가 유효한지 또는 무효한지를 나타내는 유효정보를 포함하며, 판단수단은 상기 유효정보에 근거하여 삭제된 보조데이터가 있는지 없는지를 판단하는 편집장치.

**청구항 15.** 제 14항에 있어서,

상기 제어수단은 유효한 보조데이터와 결합된 보조 관리데이터와 상기 보조데이터를 새롭게 생성된 보조 데이터 파일에 복사함으로써 불필요한 데이터를 삭제하는 편집장치.

**청구항 16.** 한 개 또는 다수의 메인데이터와 상기 메인데이터와 관련된 보조데이터중 최소한 한 개로 구성된 보조데이터 파일과 상기 메인데이터를 제어하는 관리데이터와, 상기 보조데이터를 제어하는 보조 관리데이터를 기록하는 기록매체에 기록된 보조데이터를 삭제함으로써 편집작업을 수행하고, 상기 보조데이터 파일은 상기 관리데이터와 연결 및 링크되어지는 편집장치에 있어서,

상기 기록매체로부터 상기 데이터를 판독하고, 상기 기록매체에 입력시키는 액세스수단과,

상기 액세스수단에 의해 판독된 상기 보조데이터 파일내에서 삭제되는 보조데이터와 결합된 한 개 또는 다수의 보조 관리데이터를 재입력시키는 단계와,

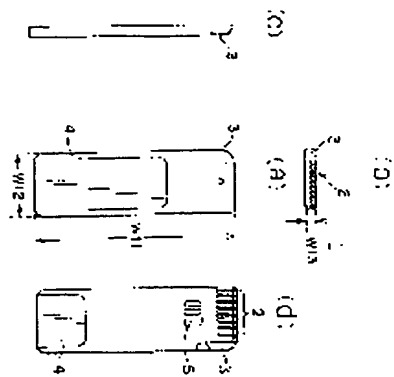
상기 기록 매체내에 재입력된 상기 명시된 한 개 또는 다수의 보조 관리데이터를 가지는 상기 보조데이터 파일을 다시 입력시키는 단계를 실행시킴으로써 상기 액세스수단을 제어하는 제어수단으로 구성된 편집장치.

**청구항 17.** 제 16항에 있어서,

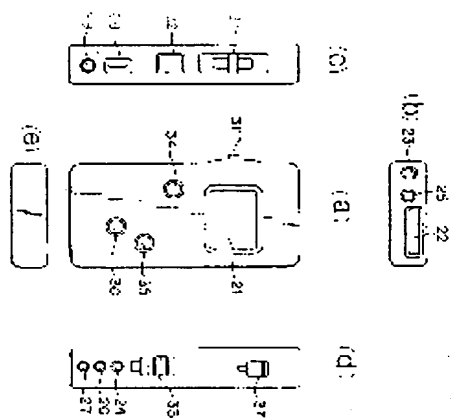
상기 보조 관리데이터는 최소한 상기 보조데이터 파일내에 저장된 보조데이터가 유효한지 또는 무효한지를 나타내는 유효 정보를 포함하며, 상기 제어수단은 상기 유효정보를 무효 상태로 설정함으로써 상기 명시된 한 개 또는 다수의 보조 관리데이터를 재입력시키는 편집장치.

도면

도면1



도면2



**도표 3**

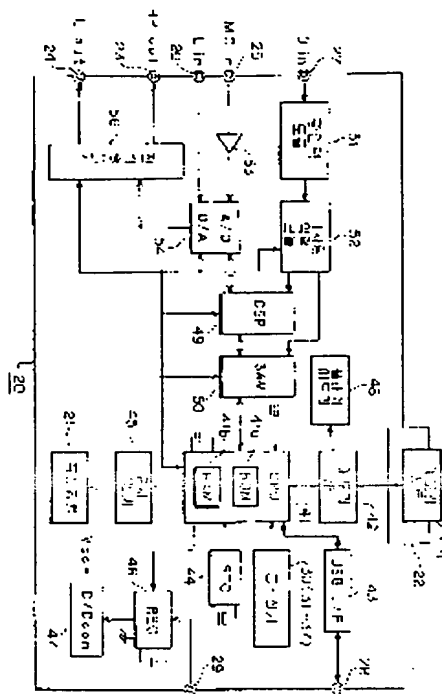
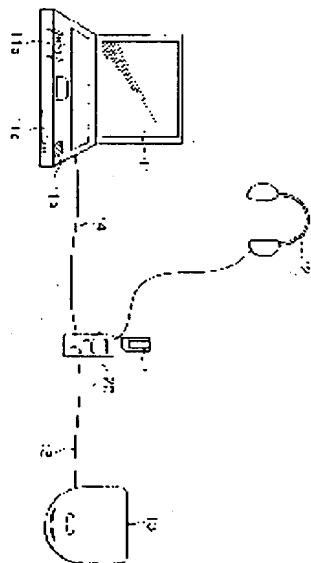


도표 4

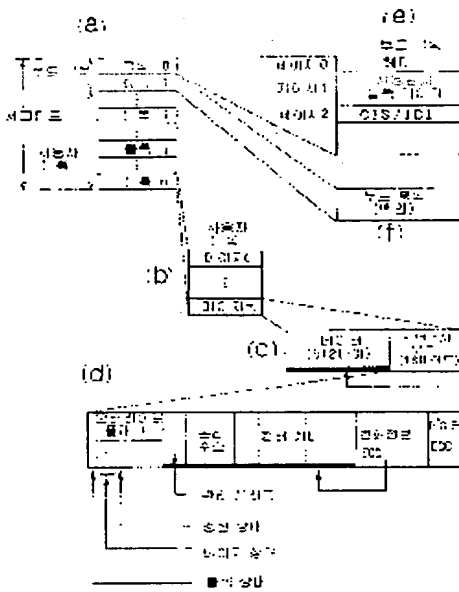


도면5

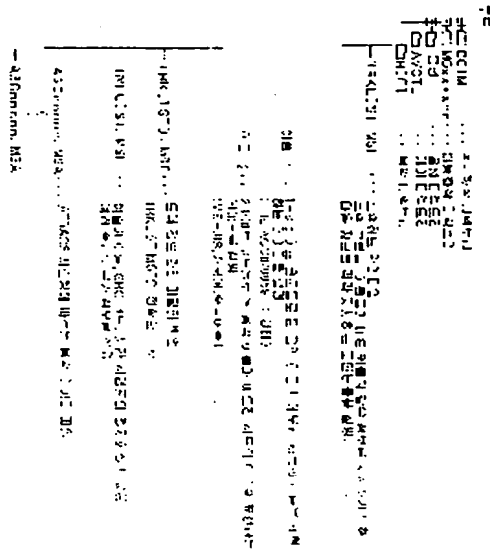
출력부
출력부 제어부
출력부 제어부
출력부 제어부
출력부 제어부

출력부 제어부 제어부

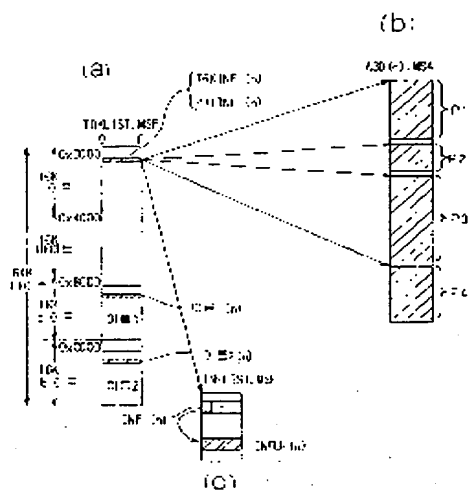
도면6



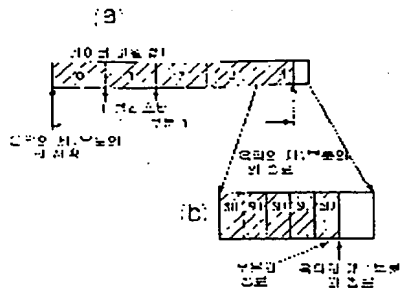
도 27



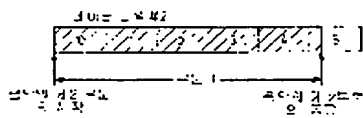
도 28



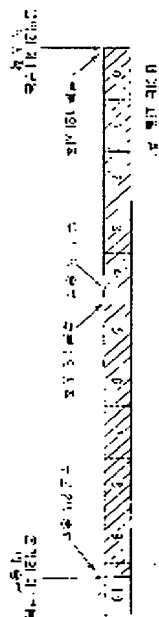
도면9



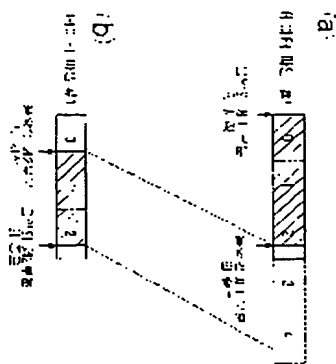
(C)



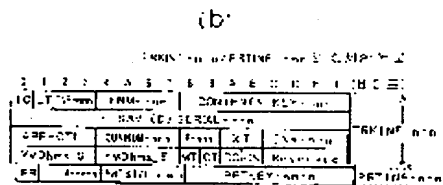
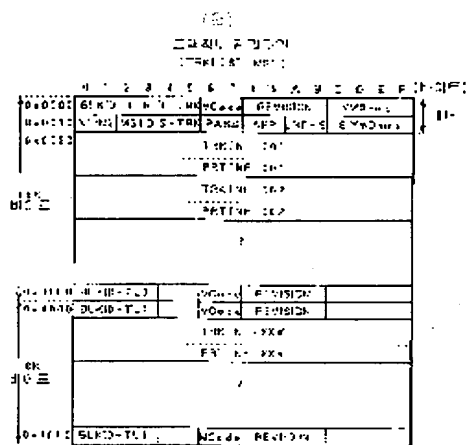
도면10



도면 11



**도면 12**











도 22

**EN23**

**도면24**

	C	2	3	4	5	6	7	8	9	A	B	C	D	E
Cx01	56					CC	DD	EE	FF	Code				
Cx02	YMDays_1	YMDays_2	YMDays_3	YMDays_4	YMDays_5	YMDays_6	YMDays_7	YMDays_8	YMDays_9	YMDays_10	YMDays_11	YMDays_12	YMDays_13	YMDays_14
Cx03	YMDays_15	YMDays_16	YMDays_17	YMDays_18	YMDays_19	YMDays_20	YMDays_21	YMDays_22	YMDays_23	YMDays_24	YMDays_25	YMDays_26	YMDays_27	YMDays_28
Cx04	YMDays_29	YMDays_30	YMDays_31	YMDays_32	YMDays_33	YMDays_34	YMDays_35	YMDays_36	YMDays_37	YMDays_38	YMDays_39	YMDays_40	YMDays_41	YMDays_42
Cx05	YMDays_43	YMDays_44	YMDays_45	YMDays_46	YMDays_47	YMDays_48	YMDays_49	YMDays_50	YMDays_51	YMDays_52	YMDays_53	YMDays_54	YMDays_55	YMDays_56
Cx06	YMDays_57	YMDays_58	YMDays_59	YMDays_60	YMDays_61	YMDays_62	YMDays_63	YMDays_64	YMDays_65	YMDays_66	YMDays_67	YMDays_68	YMDays_69	YMDays_70
Cx07	YMDays_71	YMDays_72	YMDays_73	YMDays_74	YMDays_75	YMDays_76	YMDays_77	YMDays_78	YMDays_79	YMDays_80	YMDays_81	YMDays_82	YMDays_83	YMDays_84
Cx08	YMDays_85	YMDays_86	YMDays_87	YMDays_88	YMDays_89	YMDays_90	YMDays_91	YMDays_92	YMDays_93	YMDays_94	YMDays_95	YMDays_96	YMDays_97	YMDays_98
Cx09	YMDays_99	YMDays_100	YMDays_101	YMDays_102	YMDays_103	YMDays_104	YMDays_105	YMDays_106	YMDays_107	YMDays_108	YMDays_109	YMDays_110	YMDays_111	YMDays_112
Cx10	YMDays_113	YMDays_114	YMDays_115	YMDays_116	YMDays_117	YMDays_118	YMDays_119	YMDays_120	YMDays_121	YMDays_122	YMDays_123	YMDays_124	YMDays_125	YMDays_126
Cx11	YMDays_127	YMDays_128	YMDays_129	YMDays_130	YMDays_131	YMDays_132	YMDays_133	YMDays_134	YMDays_135	YMDays_136	YMDays_137	YMDays_138	YMDays_139	YMDays_140
Cx12	YMDays_141	YMDays_142	YMDays_143	YMDays_144	YMDays_145	YMDays_146	YMDays_147	YMDays_148	YMDays_149	YMDays_150	YMDays_151	YMDays_152	YMDays_153	YMDays_154
Cx13	YMDays_155	YMDays_156	YMDays_157	YMDays_158	YMDays_159	YMDays_160	YMDays_161	YMDays_162	YMDays_163	YMDays_164	YMDays_165	YMDays_166	YMDays_167	YMDays_168
Cx14	YMDays_169	YMDays_170	YMDays_171	YMDays_172	YMDays_173	YMDays_174	YMDays_175	YMDays_176	YMDays_177	YMDays_178	YMDays_179	YMDays_180	YMDays_181	YMDays_182
Cx15	YMDays_183	YMDays_184	YMDays_185	YMDays_186	YMDays_187	YMDays_188	YMDays_189	YMDays_190	YMDays_191	YMDays_192	YMDays_193	YMDays_194	YMDays_195	YMDays_196
Cx16	YMDays_197	YMDays_198	YMDays_199	YMDays_200	YMDays_201	YMDays_202	YMDays_203	YMDays_204	YMDays_205	YMDays_206	YMDays_207	YMDays_208	YMDays_209	YMDays_210
Cx17	YMDays_211	YMDays_212	YMDays_213	YMDays_214	YMDays_215	YMDays_216	YMDays_217	YMDays_218	YMDays_219	YMDays_220	YMDays_221	YMDays_222	YMDays_223	YMDays_224
Cx18	YMDays_225	YMDays_226	YMDays_227	YMDays_228	YMDays_229	YMDays_230	YMDays_231	YMDays_232	YMDays_233	YMDays_234	YMDays_235	YMDays_236	YMDays_237	YMDays_238
Cx19	YMDays_239	YMDays_240	YMDays_241	YMDays_242	YMDays_243	YMDays_244	YMDays_245	YMDays_246	YMDays_247	YMDays_248	YMDays_249	YMDays_250	YMDays_251	YMDays_252
Cx20	YMDays_253	YMDays_254	YMDays_255	YMDays_256	YMDays_257	YMDays_258	YMDays_259	YMDays_260	YMDays_261	YMDays_262	YMDays_263	YMDays_264	YMDays_265	YMDays_266
Cx21	YMDays_267	YMDays_268	YMDays_269	YMDays_270	YMDays_271	YMDays_272	YMDays_273	YMDays_274	YMDays_275	YMDays_276	YMDays_277	YMDays_278	YMDays_279	YMDays_280
Cx22	YMDays_281													

도 25

출력 데이터 (HARDWARE) (ASCII NAME+ISRC+TCCD)

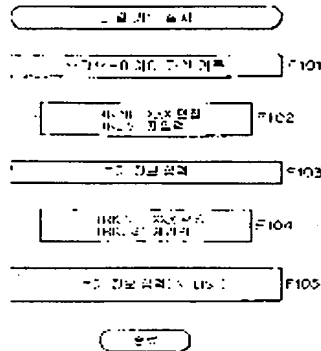
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
95					00	C2	MCODE									
96					00	C3	MCODE									
97					00	C1	MCODE									
98					00	C1	MCODE									

도 26

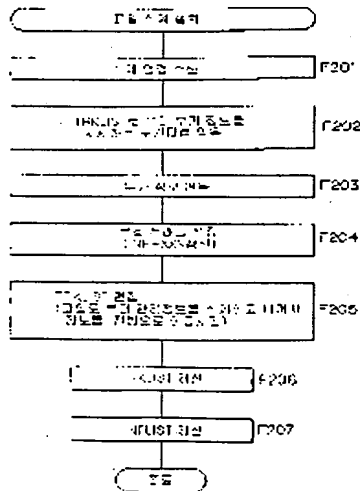
출력 데이터 (HARDWARE) (ASCII NAME+ISRC+TCCD)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
95					00	02	MCODE									
96					00	03	MCODE									
97					00	01	MCODE									
98					00	01	MCODE									

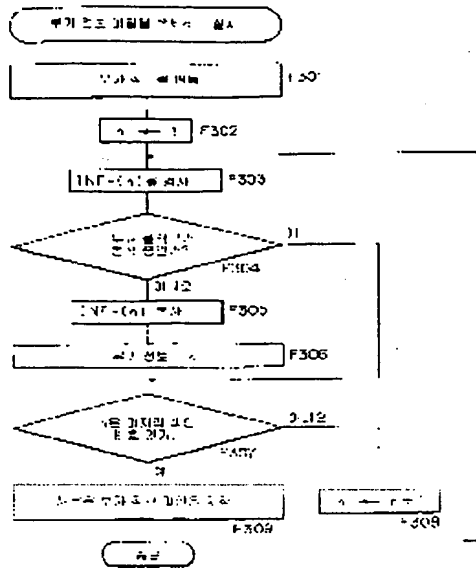
도 27



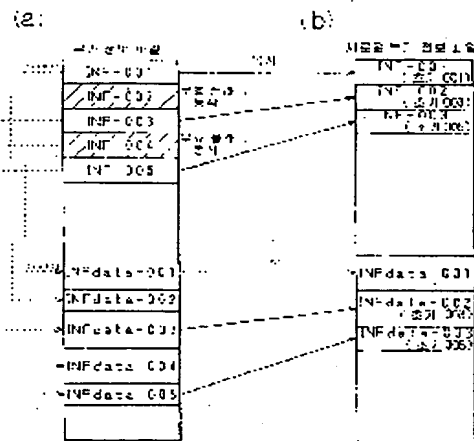
도 28



도면29



도면30



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**